



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

DESARROLLO Y PUESTA A PUNTO DE UNA APLICACIÓN WEB
PARA LA GESTIÓN DE LA HEMEROTECA DEL INSTITUTO
NACIONAL DE PATRIMONIO CULTURAL REGIONAL 3.

Trabajo de titulación presentado para optar al grado académico de:
INGENIERO EN SISTEMAS INFORMÁTICOS

AUTOR: SAÚL JAIDAN IBARRA MOREIRA.

TUTOR: DR. BYRON ERNESTO VACA BARAHONA.

Riobamba-Ecuador

2017

@2017, Saúl Jaidan Ibarra Moreira.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Saúl Jaidan Ibarra Moreira

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

El Tribunal del Trabajo de Titulación certifica que: El trabajo de investigación: **DESARROLLO Y PUESTA A PUNTO DE UNA APLICACIÓN WEB, PARA LA GESTIÓN DE LA HEMEROTECA DEL INSTITUTO NACIONAL DE PATRIMONIO CULTURAL REGIONAL 3.**, de responsabilidad del señor Saúl Jaidan Ibarra Moreira, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

FIRMA

FECHA

Ing. Washington Luna E.

**DECANO DE LA FACULTAD DE
INFORMATICA Y ELECTRONICA**

Ing. Patricio Moreno

**DIRECTOR DE LA ESCUELA DE
INGENIERÍA EN SISTEMA**

Dr. Byron Vaca Barahona

**DIRECTOR DE TRABAJO DE
TITULACIÓN**

Dr. Omar Gómez Gómez

MIEMBRO DEL TRIBUNAL

Yo, Saúl Jaidan Ibarra Moreira soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del mismo pertenece a la Escuela Superior Politécnica de Chimborazo.

Saúl Jaidan Ibarra Moreira

AGRADECIMIENTO

Agradezco a Dios, por la bendición de permitirme llegar hasta donde estoy, brindarme humildad y llenarme de valor para enfrentar todo tipo de pruebas, a mi madre Marisol y a la familia Chiluisa Cabeza por su incondicional apoyo en todas las etapas de mi vida. A mis hermanos por su paciencia y dedicación para la unión de la familia. A mi tutor Dr. Byron Vaca al Ing. Fabián Villa por la guía profesional y humana dedicada a este trabajo de titulación.

Saúl

DEDICATORIA

Dedico este trabajo a Dios por todas sus bendiciones en esta meta alcanzada. A mis padres Saúl Ibarra y Marisol Moreira por ser pilares fundamentales en mi camino, a mi familia, a la familia Chiluisa Cabeza y amigos que siempre estuvieron apoyándome y confiando en mí.

Saúl

TABLA DE CONTENIDO

AGRADECIMIENTO	iv
DEDICATORIA.....	v
TABLA DE CONTENIDO	vi
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	xi
RESUMEN	xiii
SUMMARY	xiv
INTRODUCCIÓN	1
Objetivo General.....	3
Objetivos Específicos	3
CAPÍTULO I	
1. MARCO TEÓRICO	4
1.1 Hemeroteca.....	4
1.1.1 Hemeroteca digital.....	4
1.1.2 Hemeroteca del Instituto Nacional de Patrimonio Cultural	4
1.2 Fundamentos Web	5
1.3 Aplicación Web	6
1.4 Servidor Web o Back-end	6
1.5 Apache	6
1.6 Front-end	7
1.7 Lenguaje de programación PHP	7
1.8 Frameworks PHP.....	7
1.9 Laravel	9
1.9.1 Características principales	9
1.9.2 Estructura de desarrollo	10
1.9.3 Responsabilidades, nombramiento y convenciones.....	11
1.9.4 Paradigma del MVC	12
1.9.5 Testing sobre Laravel.....	14
1.10 PHPUnit.....	15
1.11 MySQL.....	15

1.11.1	<i>Principales Características</i>	16
1.11.2	<i>Ventajas</i>	16
1.12	JSON (JavaScript Object Notation)	17
1.12.1	<i>Características</i>	17
1.12.2	<i>Ventajas</i>	17
1.13	Interfaz de Usuario	18
1.13.1	<i>Twitter Bootstrap</i>	18
1.13.2	<i>Características</i>	18
1.14	Metodología TDD	19
1.14.1	<i>Características</i>	19
1.14.2	<i>Ventajas</i>	20
1.14.3	<i>Desventajas</i>	20
CAPÍTULO II		
2.	MARCO METODOLÓGICO	21
2.1.	Fases de la Metodología	21
2.1.1.	<i>Preparación del Proyecto</i>	21
2.1.2.	<i>Personas y Roles del Proyecto</i>	22
2.1.3.	<i>Desarrollo dirigido por test de aceptación.</i>	22
2.1.4.	<i>Tipos y Roles de Usuario</i>	24
2.1.5.	<i>Recursos de hardware</i>	25
2.1.6.	<i>Planificación con TDD</i>	25
2.1.7.	<i>Desarrollo con TDD</i>	27
2.1.7.1.	<i>Historias de técnicas preliminares</i>	27
2.1.7.2.	<i>Diseño de base de datos</i>	28
2.1.7.3.	<i>Estándar de Codificación</i>	30
2.1.7.4.	<i>Desarrollo de las historias de usuario bajo TDD</i>	31
2.1.7.5.	<i>Diseño de Interfaces</i>	43
CAPÍTULO III		
3.	IMPLEMENTACIÓN Y RESULTADOS	46
3.1.	Configuración de servicios	46
3.1.1.	<i>Características físicas del servidor</i>	46
3.1.2.	<i>Instalación del sistema operativo en el servidor físico</i>	46

3.1.3.	<i>Instalación del Servidor Apache</i>	55
3.1.4.	<i>Intalación y Configuración de PHP</i>	56
3.1.5.	<i>Configuración de Selinux</i>	56
3.1.6.	<i>Configuración de directorios virtuales.....</i>	57
3.1.6.1.	<i>Crear la estructura del directorio.....</i>	57
3.1.6.2.	<i>Otorgar permisos</i>	58
3.1.6.3.	<i>Crear los archivos Virtual Host.....</i>	58
3.1.6.4.	<i>Habilitar los nuevos archivos Virtual Host</i>	60
3.1.6.5.	<i>Configurar archivos locales.....</i>	60
3.1.7.	<i>Configuración de permisos para la carpeta del proyecto</i>	60
3.1.8.	<i>Instalación de Mysql.....</i>	61
3.2.	Técnica implementada para pruebas de usabilidad	61
3.3.	Análisis de resultados.....	66
CONCLUSIONES.....		68
RECOMENDACIONES.....		69
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-2: Personas y Roles	22
Tabla 2-2: Historias de Usuario	23
Tabla 3-2: Tipos y Roles.....	25
Tabla 4-2: Product Backlog	26
Tabla 5-2: Pregunta 1.....	28
Tabla 6-2: Pregunta 2.....	28
Tabla 7-2: Pregunta 3.....	29
Tabla 8-2: Pregunta 4.....	33
Tabla 9-2: Pregunta 5.....	33
Tabla 10-2: Pregunta 6.....	33
Tabla 11-2: Pregunta 7.....	36
Tabla 12-2: Pregunta 8.....	36
Tabla 13-2: Pregunta 9.....	37
Tabla 14-2: Pregunta 10.....	37
Tabla 15-2: Pregunta 11.....	37
Tabla 16-2: Pregunta 12.....	37
Tabla 17-2: Pregunta 13.....	38
Tabla 18-2: Pregunta 14.....	38
Tabla 19-2: Pregunta 15.....	39
Tabla 20-2: Pregunta 16.....	39
Tabla 21-2: Pregunta 17.....	39
Tabla 22-2: Pregunta 18.....	40
Tabla 23-2: Pregunta 19.....	40
Tabla 24-2: Pregunta 19.....	40
Tabla 25-2: Pregunta 19.....	41
Tabla 26-2: Pregunta 20.....	41
Tabla 27-2: Pregunta 21.....	42
Tabla 28-2: Pregunta 22.....	42

Tabla 29-2: Pregunta 23.....	42
Tabla 30-2: Pregunta 24.....	43
Tabla 31-3: Esquema de evaluación	62
Tabla 32-3: Partes de la encuesta de Shneiderman.....	63
Tabla 33-3: Reacciones globales del sistema	64
Tabla 34-3: Pantallas	64
Tabla 35-3: Terminología e información del sistema.....	65
Tabla 36-3: Aprendizaje	65
Tabla 36-3: Capacidad del sistema	66

ÍNDICE DE FIGURAS

Figura 1-1: Request y Response de cliente a servidor web	5
Figura 2.1: Logo de Laravel	9
Figura 3.1: Estructura de un proyecto en Laravel	10
Figura 4-1: Patrón de diseño.....	12
Figura 5-1: Marco lógico de trabajo de la arquitectura MVC.	13
Figura 6-1: Object en un objeto JSON.	17
Figura 7-1: Array en un objeto JSON.....	17
Figura 8-1: Fases Metodología TDD.....	19
Figura 9-2: Modelo físico de la BD.....	30
Figura 10-2: Estructura del proyecto - Hemeroteca	31
Figura 11-2: Interfaz de inicio de sesión	44
Figura 12-2: Interfaz para el menu de usuario.....	44
Figura 13-2: Interfaz para la gestión de obras	45
Figura 14-2: Interfaz para reportes	45
Figura 15-3: Selección del sistema operativo.....	46
Figura 16-3: Selección del idioma.....	47
Figura 17-3: Opciones de instalación	47
Figura 18-3: Escoger zona horaria	48
Figura 19-3: Configuración del teclado.....	48
Figura 20-3: Selección del idioma del sistema.....	49
Figura 21-3: Tipo de instalación del sistema operativo.....	49
Figura 22-3: Confihuración de las particiones	50
Figura 23-3: Montaje de las particiones	50
Figura 24-3: Nuevo punto de montaje.....	51
Figura 25-3: Asignación de espacio en disco a las particiones	51
Figura 26-3: Configuración de la red del servidor	52
Figura 27-3: Configuración de la IP estática del servidor	52
Figura 28-3: Inicio de la instalación.....	53

Figura 29-3: Creación de los usuarios del sistema	53
Figura 30-3: Contraseña del root	54
Figura 31-3: Instalación completa	54
Figura 32-3: Escritorio Centos 7	55
Figura 33-3: Instalación Apache	55
Figura 34-3: Comandos de configuración Apache	55
Figura 35-3: Comprobación instalación Apache	56
Figura 36-3: Configuración de selinux, seguridad del kernel	57
Figura 37-3: Verificar la configuración de selinux	57

RESUMEN

La presente investigación tuvo como objetivo desarrollar una aplicación web para la gestión documental de la Hemeroteca del Instituto Nacional de Patrimonio Cultural (INPC) regional 3, con sede en la ciudad de Riobamba provincia de Chimborazo. El análisis y uso de herramientas de desarrollo web, del gestor de base de datos MySQL conjuntamente con el framework Laravel, permitieron determinar que poseen un alto nivel de compatibilidad y desempeño, por lo que se consideraron ideales para el procesamiento de la información en tiempo real, permitiendo obtener productos funcionales, aceptables y fáciles de usar para el usuario final. Durante el proceso del desarrollo del sistema se utilizaron los componentes del framework laravel como Blade y Eloquent componentes principales y de gran versatilidad al momento de la programación en esta herramienta Por todas las características y ventajas mencionadas se obtiene un producto final que se encuentra en funcionamiento en el INPC cumpliendo con todos los requerimientos establecidos por el cliente. La aplicación web aceptada en el 100% es el resultado de haber realizado un desarrollo de software aplicando las nuevas metodologías de programación, como es desarrollo guiado por pruebas (TDD) para “HEMEROTECA VIRTUAL INPC”. Se concluyó que la aplicación permite generar reportes de las obras prestadas con óptimos tiempos de proceso. Se recomienda el estudio minucioso y una comprensión técnica avanzada del funcionamiento de la aplicación en cuanto a su integración con otras herramientas de trabajo.

PALABRAS CLAVE: <TECNOLOGIA Y CIENCIAS DE LA INGENIERIA>, <SISTEMAS INFORMÁTICOS>, <APLICACIÓN WEB>, < MYSQL (MOTOR DE BASE DE DATOS)>, <LARAVEL(FRAMEWORK)>, <BLADE>, <ELOQUENT>, <DESARROLLO GUIADO POR PRUEBAS (TDD)>.

SUMMARY

The aim of this research is to develop a web application for document management at the newspaper and periodicals library of Instituto Nacional de Patrimonio Cultural (INPC) Regional 3 in the headquarters of Riobamba, province of Chimborazo. Having analyzed and used MySQL and laravel, it was possible to determine they were highly compatible and their performance was high. That is why we believe they are ideal for real time data process obtaining a simple and acceptable product for users. The main Laravel, Blade and Eloquent components were used when programming this tool during the system development process. Therefore, a final product was gotten, which is being applied in INPC meeting all the requirements established by the customers. The web application accepted 100% is the result of the software development using new programming technologies such as test driven development (TDD) for “VIRTUAL NEWSPAPER AND PERIODICALS LIBRARY INCP”. We concluded that borrowed document reports were gotten on time. We recommend to study carefully and understand technically the application compatibility with other tools.

KEYWORDS: <ENGINEERING SCIENCE AND TECHNOLOGY>, <TECHNOLOGY SYSTEMS>, <WEB APPLICATION>, <MySQL (DATABASE ENGINE)>, <LARAVEL(FRAMEWORK)>, <BLADE, ELOQUENT>, <TEST DRIVEN DEVELOPMENT (TDD)>

INTRODUCCIÓN

La Constitución de la República del Ecuador promueve la reorganización del estado integrando a los sectores estratégicos del Ecuador, el Plan Nacional para el Buen Vivir se basa en las políticas establecidas en la Constitución, en ella se encuentra en ejecución el Plan Nacional de desarrollo para el período 2013-2017 (Plan Nacional Buen Vivir, 2013).

El onceavo objetivo del plan da sustento al desarrollo del trabajo de titulación, el mismo expresa “Asegurar la soberanía y eficiencia de los sectores estratégicos para la transformación industrial y tecnológica” (Plan Nacional Buen Vivir, 2013).

En su inciso de Políticas y lineamientos se denota, “Democratizar la prestación de servicios públicos de telecomunicaciones y de tecnologías de información y comunicación (TIC), incluyendo radiodifusión, televisión y espectro radioeléctrico, y profundizar su uso y acceso universal” (Plan Nacional Buen Vivir, 2013).

Las alternativas de gestionar información por medio de las TIC son varias, unas de ellas es la utilización de aplicaciones web, donde las instituciones públicas como privadas ven la forma de actualizarse en el mundo moderno permitiendo el acceso a la información con una disponibilidad total desde cualquier lugar donde el acceso de usuarios es simultáneamente sin necesidad de estar presente físicamente. (mora, 2002)

El Instituto Nacional de Patrimonio Cultural (INPC) es una entidad del sector público con alcance nacional, y tiene como misión investigar, normar, regular, asesorar y promocionar las políticas sectoriales para la gestión patrimonial en miras a la preservación, conservación, apropiación y uso adecuado del patrimonio tangible e intangible del Ecuador (INPC, 2015).

Uno de los principales objetivos estratégicos del Instituto se refiere en diseñar e implementar sistemas de información para la gestión de los bienes patrimoniales y culturales.

El INPC tiene como fin la recopilación de la documentación producida en la institución a nivel nacional, tanto histórica como actual. Toda la información que genere esta base de datos, servirá para la GESTION DEL PATRIMONIO sobre todo para los Gobiernos Autónomos Descentralizados GAD's, cuyo mandato constitucional ahora es total.

En representación del Instituto Nacional de Patrimonio Cultural Regional 3, su directora afirma que la recopilación de la documentación producida en la Institución, tanto histórica como actual la realiza en forma personal, es decir, sus obras literarias se obtienen a través de donaciones del sector público como privado e inversiones económicas para su adquisición.

Los antecedentes planteados, permite considerar al Instituto Nacional de Patrimonio Cultural como ente para la innovación tecnológica en sus procesos institucionales, esta iniciativa impulsará la calidad, seguridad y cobertura en la prestación de servicios públicos, a través del uso de las TIC's, considerado en el objetivo onceavo del Plan del Buen Vivir.

Las obras literarias obtenidas reposan en la hemeroteca de la misma institución en manera física, con gestión y control de sus bienes en forma manual, llevando un proceso sin mayor control y seguimiento, este puede considerarse un proceso deficitario sin promover la eficiencia en la administración de los recursos limitados como son las obras literarias producidas en el país, concretamente de la provincia de Chimborazo.

El levantamiento de información inicialmente realizado, indica una problemática en la gestión y control por parte de la INPC en sus bienes patrimoniales. La dificultad de organización, transparencia, consulta, difusión e inclusión social del patrimonio bibliográfico es causada por el precario seguimiento de inventariado que se le presta a las mismas.

Se identifica o deriva problemas de pérdida de obras literarias, altos costos de seguimiento, individualización, recuperación, preservación y conservación de obras rescatadas que no fueron debidamente identificadas en el servicio de prestación de las obras. Las inversiones económicas a nuevos ejemplares no inventariados hacen de un servicio público no óptimo, ya que limita la exposición de la identidad y valor social que tiene la literatura ecuatoriana.

Se pone en consideración. ¿Cómo influye la implementación de una aplicación web para la gestión y control de la hemeroteca en el proceso de prestaciones, donaciones y adquisiciones de obras que se va realiza en el Instituto Nacional de Patrimonio Cultural Regional 3?

Esta interrogante y problemática converge en la propuesta de desarrollo y puesta a punto de una aplicación Web para la gestión de la Hemeroteca del Instituto Nacional de Patrimonio Cultural

Regional 3 (INPC), con un seguimiento metodológico ágil de desarrollo software bajo los lineamientos de la aplicación del Desarrollo Dirigidos por Pruebas (TDD).

En la propuesta de desarrollo de la aplicación web, se plantea objetivos que presidirán para cumplir con el propósito del trabajo de titulación.

Objetivo General

- Desarrollar y poner a punto una aplicación web basado en el framework Laravel con el gestor de bases de datos MySQL, para la gestión de la hemeroteca del Instituto Nacional de Patrimonio Cultural Regional 3.

Objetivos Específicos

- Estudiar y comprender el Framework de programación web Laravel y el gestor de bases de datos MySQL para desarrollar el aplicativo web.
- Desarrollar la aplicación web con la metodología TDD para cada requerimiento del INPC.
- Configurar todos los servicios necesarios para el funcionamiento de la aplicación web
- Realizar la respectiva prueba de aceptación de la aplicación web.

Los objetivos planteados marcan las pautas del desarrollo técnico: el modelo de proceso o el marco metodológico a cumplir, que permiten finalizar con el producto software totalmente operativo y su implementación según los requerimientos planteados.

El sistema web “**HEMEROTECA VIRTUAL INPC**” entre sus funcionalidades más importante contempla llevar el control del inventario de obras literarias, así como gestión y seguimiento de prestaciones.

Se resalta la importancia de reportes para el seguimiento y cobertura en la prestación del servicio público facilitando las obras a quien necesite, esto influye en la identificación, recuperación, preservación y conservación del patrimonio asegurando la inclusión social para toda la ciudadanía que requiera del servicio.

CAPÍTULO I

1. MARCO TEÓRICO

El Instituto Nacional de Patrimonio Cultural (INPC) se consolida como un centro de investigación especializado, con capacidad científico-técnica, que permita el estudio, análisis y aplicación de teorías, metodologías y técnicas, para: catalogar, documentar, proteger y potenciar los bienes patrimoniales, con la finalidad de difundir y lograr la concienciación de los diversos actores involucrados, sobre la importancia y preservación del patrimonio cultural para beneficio de las presentes y futuras generaciones. (INPC, 2015).

Una de las funciones de la Hemeroteca del INPC regional 3 es la consulta y difusión del patrimonio bibliográfico de la región sierra sede Riobamba – Ecuador, creada mediante decreto supremo 2600 del 9 de junio del 1978. (INPC, 2015).

1.1 Hemeroteca

Es un sitio destinado para conservar, ordenar y clasificar diferentes instancias bibliográficas como libros, revistas, diarios y diferentes escritos destinados para su consulta en cualquier momento por el público como un servicio para dar a conocer el patrimonio bibliográfico ecuatoriano.

1.1.1 *Hemeroteca digital*

Consulta y difusión del patrimonio bibliográfico mediante el uso de las TIC, esto se da bajo la necesidad de la inclusión en todos los niveles de la sociedad comúnmente lograda a través del internet.

El acceso a consultas de los distintos ejemplares mediante sistemas web en línea, es un servicio que llegará a más personas consiguiendo abarcar a una gran parte de la sociedad.

1.1.2 *Hemeroteca del Instituto Nacional de Patrimonio Cultural*

La hemeroteca del INPC se centra en obtener, recopilar la documentación producida en la Institución a nivel nacional, tanto histórica como actual, difundir el patrimonio bibliográfico que se encuentra a

disposición y velar por el debido uso para la región 3, sede Riobamba, así como para la sociedad ecuatoriana en general. (INPC, 2015).

El sistema informático orientado a la web propuesta, pretende impulsar y ayudar en la exploración rápida, análisis y participación integral de la información referida en todas las áreas que conformen las obras literarias. A través de esta herramienta se consigue acceso integrado a literatura de alta calidad por medio de una plataforma de acceso universal con extensa variedad de contenido bibliográfico como patrimonio del Ecuador (INPC, 2015).

1.2 Fundamentos Web

El fundamento de la web apunta en sus dos fortalezas, el protocolo HTTP (Hypertext Transfer Protocol o en español protocolo de transferencia de hipertexto) y el lenguaje semántico HTML (HyperText Markup Language o lenguaje de marcas de hipertexto), siendo el protocolo base de la www (World Wide Web o red informática mundial) para la comunicación enriquecida entre diferentes tipos de terminales; la representación de contenido estructurado y flexible para enriquecerse de forma dinámica es representada por la figura 1-1.

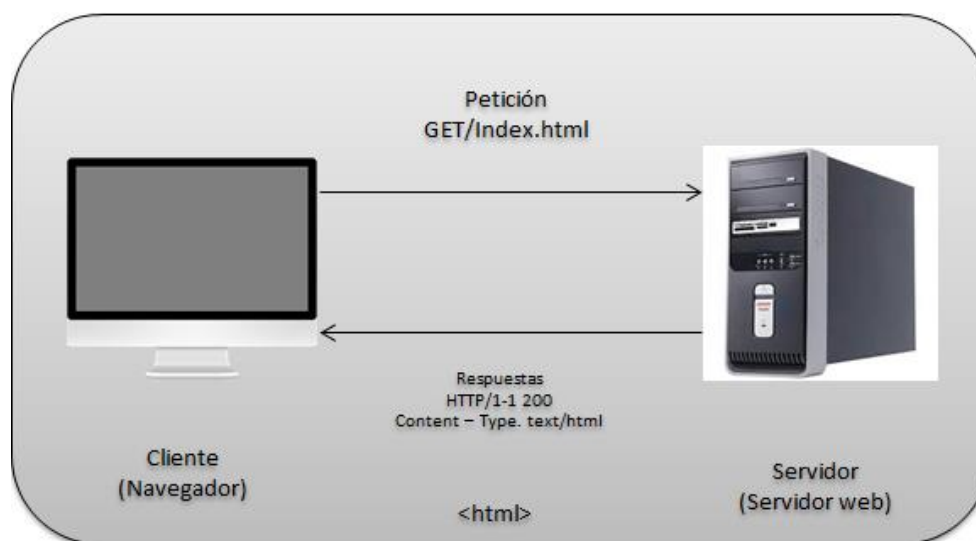


Figura 1-1: Request y Response de cliente a servidor web.

Fuente: (Mateu, 2004)

Se evidencia de forma esquemática, el modelo de procesamiento de estas dos fortalezas de la web, describiendo que, el cliente requiere una petición a una dirección y puerto establecido con algunos

datos informativos como cabeceras, y de igual forma el servidor responde con los datos solicitados por la misma dirección.

1.3 Aplicación Web

Históricamente la web era una colección de páginas estáticas que demostraba una sola información en el tiempo, ahora las aplicaciones o sistemas web no son instalados y mucho menos sobre una plataforma o sistema operativo en particular, se alojan en un servidor interconectado a la Internet o una red local; su entorno de trabajo es mucho más complicado y con resultados enriquecidos a la vista del usuario final con funcionalidades potentes y resultados personalizados. (Mateu, 2004)

Los sistemas Web se pueden utilizar en cualquier navegador para este propósito (Google Chrome, Mozilla Firefox, Internet Explorer, Opera, Zafarí) sin importar el sistema operativo, para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el mismo. (Mateu, 2004)

1.4 Servidor Web o Back-end

Un servidor web o back-end es un sistema que recibe y responde a peticiones de usuarios finales a través de navegadores que utilizan el protocolo de comunicación HTTP o HTTPS (La versión segura, cifrada y autenticada de http). Este proceso básicamente atiende a solicitudes en el puerto TCP (Protocolo de Control de Transmisión) designado, luego busca solicitudes en la cadena “request” enviando el recurso por la misma conexión. (Mateu, 2004)

Este proceso es básico de todos los servidores para presentar información de un recurso estático. Sin embargo, en la actualidad uno de los recursos más importantes de un servidor web es mostrar contenido dinámico y personalizado. Este nivel de soporte es el que se propone en el trabajo técnico de titulación. (Mateu, 2004)

1.5 Apache

Apache es un servidor web de código libre robusto cuya implementación se realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando internet

y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada. (Mateu, 2004)

Estos voluntarios se conocen como el Apache Group, además cientos de personas han contribuido al proyecto con código, ideas y documentación. Existen multitud de servidores HTTP de código libre, pero casi todos ellos han quedado eclipsados por la fama de Apache, sin embargo, algunos de estos servidores presentan características que les dotan de gran interés (Mateu, 2004)

1.6 Front-end

Es la parte del aplicativo que va a interactuar con los usuarios en la navegación y hacer peticiones contra el servidor. Esto se traduce más técnicamente en maquetar la estructura semántica del contenido en HTML, codificar el diseño en hojas de estilo con CSS (Hojas de Estilo en Cascada); y agregar la interacción con el usuario a base de lenguajes Scripting como JavaScript y Ajax.

1.7 Lenguaje de programación PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser embebido en HTML. Lo que distingue a PHP de algo del lado del cliente como Javascript, es que el código es ejecutado en el servidor, generando HTML y enviando al cliente. (Cuellen, 2002)

El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente, el servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene escrito en líneas de código.

PHP y su simplicidad, ofrece muchas características avanzadas para los programadores profesionales y no profesionales. (php, 2016)

1.8 Frameworks PHP

Desarrollar proyectos más estables, con buenas prácticas y en menor tiempo se lo hace utilizando framework que facilite obtener aplicaciones escalables, con garantías de mantenimiento porque cumple con estándares y buenas prácticas de desarrollo resolviendo código, ordenado, influyendo en resolver problemas comunes a tiempo.

Alguno de los frameworks más reconocidos en la actualidad se describe para evidenciar la estabilidad y madurez del lenguaje de programación PHP.

- **Laravel:** Es un framework PHP para desarrollar aplicaciones web con una sintaxis orientada a objetos, elegante, con tareas comunes automatizadas como autenticación, rutas, sesiones, entre otras. Este provee accesibilidad, herramientas desarrolladas en poco tiempo poderosas y robustas.
- **CodeIgniter:** Es un poderoso framework PHP para construir aplicaciones web con herramientas simples, requiere configuraciones mínimas, con reglas restrictivas flexibles y tiene características que enriquecen su entorno con librerías para tareas comunes, así como también una simple interfaz y estructura lógica para acceder a estas. Su core inicial solo necesita de unas cuantas librerías.
- **CakePHP:** Este framework está pensado para construir aplicaciones orientadas a la web de manera muy ágil y rápida, usando características de generación de código para construir prototipos de manera ligera. No hay complicación en las configuraciones iniciales, ya que su documentación manifiesta que basta la configuración de la base de datos y ya está listo para realizar desarrollo. Esto se lo realiza todo en base a las convenciones de la arquitectura MVC.
- **Symfony:** Es un framework ligero de PHP para la creación y mantenimiento en el tiempo, de aplicaciones web, usando estándares de facto de php, tal como PHPUnit, y generando nombres convencionales de desarrollo de clases. El marco de trabajo no confina por completo a su entorno.

Provee de un conjunto de componentes prefabricados que se puede integrar rápidamente en las aplicaciones en desarrollo, consiguiendo un trabajo eficiente y eficaz mediante una metodología clara facilitado por la misma. (Bean, 2015)

1.9 Laravel

Laravel es un framework que permite la creación y desarrollo de aplicaciones web, el lenguaje de programación que se utiliza en este framework es PHP, laravel está representado con el logo que muestra la figura 2-1.



Figura 2-1: Logo de Laravel.
Fuente: hechoenlaravel.com

Los framework de PHP no son nuevos, pero uno de los recientes en este lenguaje de programación es **Laravel** que ha explotado de popularidad desde su versión 3, siendo uno de los framework ampliamente utilizados en un tiempo de vida relativamente corto, este desarrollo se encuentra en los repositorios de Github del mismo nombre, Larave.l (Bean, 2015)

Es un framework que permite el uso de una sintaxis elegante y expresiva para crear código de forma sencilla y permitiendo multitud de funcionalidades, donde intenta aprovechar lo mejor de otros frameworks y las características de las últimas versiones de PHP. (Ramses, 2016)

Gran parte de Laravel está formado por dependencias, especialmente de Symfony, esto implica que el desarrollo de Laravel dependa también de la implementación de sus dependencias. (Bean, 2015)

1.9.1 *Características principales*

- Sistema de ruteo, también RESTful.
- Blade, Motor de plantillas.
- Peticiones Fluent.
- Eloquent ORM.
- Basado en Composer.
- Soporte para el caché.

- Soporte para MVC.
- Usa componentes de Symfony.

1.9.2 Estructura de desarrollo.

La estructura de desarrollo de un proyecto creado con Laravel, se contempla un directorio donde se encuentra todos los componentes y archivos de configuración del proyecto creado, como se muestra en la figura 3-1.

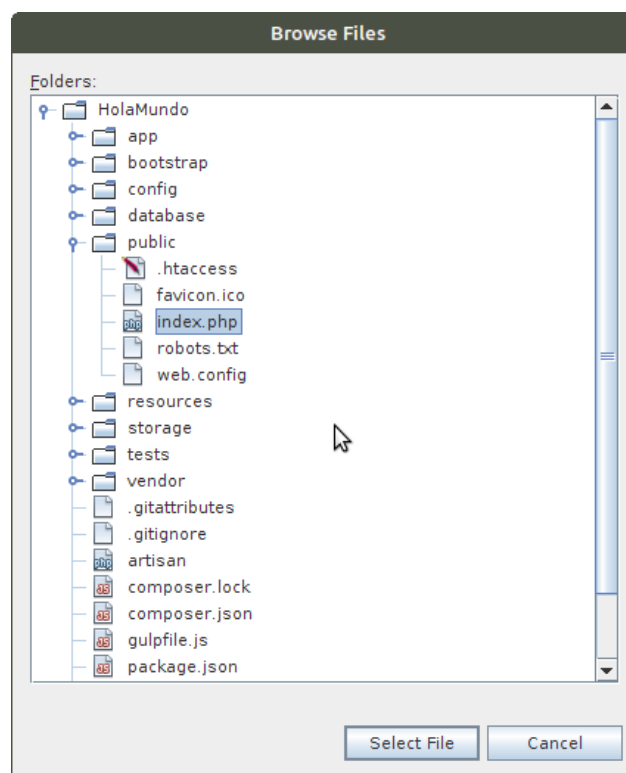


Figura 3-1: Estructura de un proyecto en Laravel.

Fuente: hechoenlaravel.com

/app: Contiene los controladores, modelos, vistas y configuraciones de la aplicación. En esta carpeta se escribe la mayoría del código para que la aplicación funcione.

/public: Es la única carpeta a la que los usuarios de la aplicación pueden acceder. Todas las peticiones y solicitudes a la aplicación pasan por esta carpeta, ya que en ella se encuentra el `index.php`, este archivo es el que inicia todo el proceso de ejecución del framework. En este directorio también se alojan los archivos CSS, JavaScript, imágenes y otros archivos que se quieran hacer públicos.

/vendor: En esta carpeta se alojan todas las librerías que conforman el framework y sus dependencias.

/lang: En esta carpeta se guardan archivos PHP que contienen Arrays con los textos de diferentes lenguajes, en caso que se desee que la aplicación se pueda traducir.

/app/config: Aquí se pueden encontrar todos los archivos de configuración de la aplicación. Base datos, cache, correos, sesiones o cualquier otra configuración general de la aplicación.

/app/controller: Contiene todos los archivos con las clases de los controladores que sirven para interactuar con los modelos, las vistas y manejar la lógica de la aplicación.

/app/model: Los modelos son clases que representan la información (datos) de la aplicación y las reglas para manipularlos. En la mayoría de los casos cada tabla de la base datos está representada con un modelo. La lógica de negocio de la aplicación está contenida en los modelos.

/app/views: Este directorio contiene las plantillas de HTML que usan los controladores para mostrar la información. Hay que tener en cuenta que en esta carpeta no van los Javascript, CSS o imágenes, ese tipo de archivos van alojados en la carpeta/public. (Ramses, 2016)

1.9.3 Responsabilidades, nombramiento y convenciones

Una de las acotaciones principales con las aplicaciones estándares realizadas sobre PHP es el lack de separación, limpia de la lógica de negocio, interfaz de usuario y el modelo consultado, comúnmente conocido como arquitectura MVC (modelo vista controlador).

El scaffolding de Laravel tiene una estructura con directorios y archivos específicos que ayudan a la toma de decisiones a veces triviales pero engorrosas para los desarrolladores; es decir, archivos con nombres específicos, estos con funciones que denotan un comportamiento definido de la instancia que la invoca, sin embargo, sus nombres están prefijados.

Así mismo ahorra tiempo en tener nombres bien definidos para las variables, métodos y tablas de la base de datos. Esto es un aporte al tiempo de implementación de nuevos CRUD (acrónimo de "Crear, Leer, Actualizar y Borrar" del original en inglés: Create, Read, Update and Delete), con un automatizado de tareas que genera rutas automáticas, al tener un estándar y convenciones de nombres para generar sin problemas nuevos archivos genéricos. (Bean, 2015)

1.9.4 Paradigma del MVC

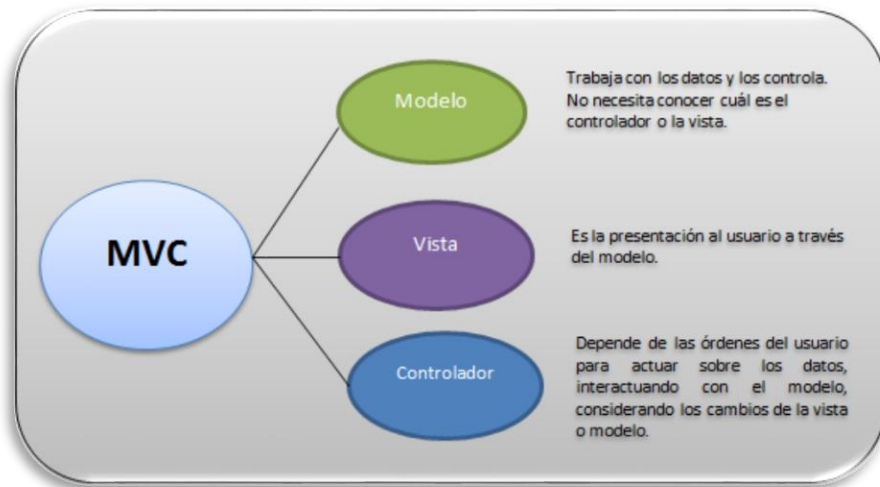


Figura 4-1: Patrón de diseño.
Realizado Por: Saúl Ibarra. 2016

La figura 4-1 muestra el patrón de arquitectura MVC que surge con el objetivo de disminuir el esfuerzo que realiza el programador, permitiendo la implementación de sistemas múltiples mediante la estandarización del diseño de la aplicación. El patrón MVC es un paradigma que descompone una aplicación en tres partes que son Modelo, Vista y Controlador. Cada parte puede ser implementada por separado, garantizando escalabilidad y permitiendo realizar mantenimiento en cortos períodos de tiempo. (Fernández Romero & Díaz González, 2012)

Modelo: Los recursos de una aplicación están representados por los modelos, esto corresponde a registros almacenados en la base de datos o datos obtenidos de alguna fuente de almacenamiento de los mismos, esto representa una entidad en la aplicación, es decir, un usuario, un artículo, una obra literaria e incluso un evento. (Fernández Romero & Díaz González, 2012)

En Laravel se representa a los modelos en clases y son nombrados automáticamente con el estándar de codificación CamelCase, se corresponde con el mismo nombre dado a la tabla de la base de datos de la entidad o modelo en cuestión.

Controladores o controladores de rutas: Este tiene una función que es, el de responder a una solicitud, resolver un proceso lógico en particular y devolver los datos procesados apropiados. Los controladores son los archivos donde se procesa los datos obtenidos, sean de una petición enviada

mediante get o post o una consulta de datos de una base de datos; para luego devolver a la vista o guardarlos en otro esquema de almacenamiento de datos. (Fernández Romero & Díaz González, 2012)

Sin embargo, tiene la flexibilidad para generar propios recursos y no es una camisa de fuerza para acatar nombres estandarizados que genera un controlador por defecto, a esto se denomina Controladores RESTful.

Un controlador RESTful permite definir propias acciones y que los métodos HTTP respondan a estos.

Vista o Template: Son responsables de mostrar la respuesta retornada desde el controlador en un formato conveniente, usualmente una página web HTML.

Es conveniente construir vistas en Laravel bajo el motor de vistas llamado “Blade template language”, pero no es estrictamente necesario, también se lo puede hacer con lenguaje PHP en su forma pura.

La extensión que toma los archivos creados con el motor generador de vistas es “.blade.php”, o en su defecto simplemente “.php”. Los hilos ejecutados por Laravel se verifican para conocer si es una vista blade o no. (Bean, 2015)

La Figura 5-1, ilustra la interacción entre todos los componentes en una aplicación web desarrollada con Laravel.

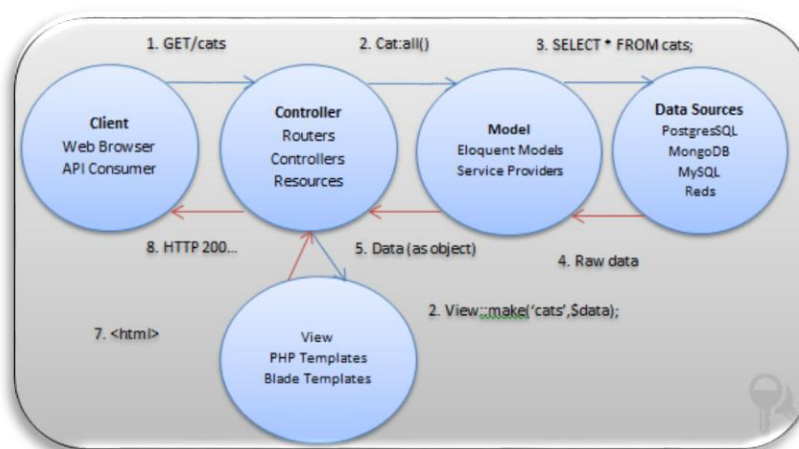


Figura 5-1: Marco lógico de trabajo de la arquitectura MVC.

Fuente: (Bean, 2015)

1.9.5 *Testing sobre Laravel*

El testing es una parte muy descuidada en el desarrollo de sistemas web hechos con PHP, esto en comparación con lenguajes de programación como Java y Ruby, donde el testing está fuertemente ligado en el flujo de trabajo de implementación. Las aplicaciones PHP tienden a ser vigorosamente acoplados y son difíciles de testear. (Bean, 2015)

Sin embargo, gracias a los esfuerzos de estandarización y modularización con marcos de trabajo que fomenta la separación de las responsabilidades en capas, las pruebas de PHP se han vuelto más accesibles y la mentalidad hacia ella está cambiando lentamente. (Bean, 2015)

Laravel fue construido desde cero para facilitar el testeo con todos los archivos necesarios para empezar, junto con diferentes auxiliares para poner a prueba su aplicación, lo que ayuda a los principiantes a superar algunos de los obstáculos más grandes. (Bean, 2015)

Beneficios del Testing

Los beneficios de hacer pruebas unitarias solo se evidencian una vez realizadas, su preparación y ejecución tienen una inversión considerable de tiempo en prototipos de corta duración, que pueden incluso parecer una completa pérdida de tiempo. (Bean, 2015)

Sin embargo, en casi todos los demás casos, cuando es probable que un proyecto crezca en complejidad, o cuando es de colaborar con otros desarrolladores, las pruebas tienen el potencial de ahorrar a otra persona una gran cantidad de tiempo. (Bean, 2015)

Si es comprobado que procesos implementados funcionan, entonces se asegura rápidamente que se sigue trabajando como se esperaba, incluso si se cambia el código fuente en una fecha posterior de la liberación del sprint implementado. Además, que obliga a definir claramente y sin ambigüedad el comportamiento esperado de su aplicación y, por tanto, pueden complementar o sustituir una parte significativa de la documentación. (Bean, 2015)

Esto puede ser particularmente útil, tanto para nuevos desarrolladores o para profesionales que tienen amplia experiencia en la implementación de software (Bean, 2015).

1.10 PHPUnit

Un efecto positivo de las pruebas o testing, es que obliga a dividir el código en dependencias pequeñas y manejables, de modo que se puede probar de forma aislada a las clases y métodos individuales, esto se conoce como pruebas unitarias. Se basa en un marco de pruebas denominados PHPUnit, que distingue un gran número de herramientas para la creación de prueba pequeñas. (Bean, 2015)

Un gran aporte para aprender diferentes formas en que se puede probar o testear, es mirar cómo sus autores han escrito pruebas para ello.

Ahora, cuando se está seguro de que todas las clases interdependientes funcionan, se lleva a cabo otro tipo de pruebas. Se propone simular el tipo de interacción que un usuario tendría a través de un navegador web, este usuario haría, por ejemplo, visitar una URL específica, realizar ciertas acciones, y espera ver algún tipo de retroalimentación de la aplicación. (Bean, 2015)

Es el tipo de prueba más sencillo, de forma manual se realiza cada vez que se actualiza el navegador al generar cambios en el código. Al inicio de un proyecto es conveniente realizar este tipo de pruebas. Sin embargo, hay que tener en cuenta que, si se produce algún error, se tendrá que analizar y revisar el código para encontrar el componente exacto que provocó dicho error. (Bean, 2015)

1.11 MySQL

Es un sistema de gestión de bases de datos relacional, creado por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece y en la posibilidad de integrar este gestor en un software propietario, ya que, de otra manera, se vulneraría la licencia GPL. (Corp, 2002)

El lenguaje de programación que utiliza Mysql es Structured Query Language (SQL) desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales, permitiendo a los usuarios y desarrolladores disponer recursos de administración de datos para poder agregar, acceder o procesar esta información guardada en el ordenador, y esta es la función que realiza Mysql.(Corp, 2002)

1.11.1 Principales Características

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas: Numéricos, de fechas, de cadena.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos (Linux/Windows – Apache- MySQL- PHP/Perl/Python). MySQL funciona sobre múltiples plataformas, incluyendo:
AIX, BSD, FreeBSD, HP-UX, GNU/Linux, Mac OS X, NetBSD, Novell Netware, OpenBSD, OS/2 Warp, QNX, SGI IRIX, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64, EBD.
- Cada base de datos cuenta con 3 archivos: uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
Para cada base de datos creada, MySQL crea un directorio con el nombre asignado a la base de datos. Dentro de este directorio, por cada tabla que definida MySQL va a crear tres archivos: mitabla.ISD, mitabla.ISM, mitabla.frm.
- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas. (Corp, 2002)

1.11.2 Ventajas

- Velocidad al realizar las operaciones SQL, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de Sistemas Operativos.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Conectividad y seguridad. (Goretty, 2015)

1.12 JSON (JavaScript Object Notation)

Desde su página oficial JSON se describe como un formato ligero y práctico para el intercambio de datos, tanto para el usuario es de fácil manipulación y comprensión como para un terminal es sencillo interpretarlo y generarlo. (*json.org, 2016*)

JSON es un formato de texto completamente independiente del lenguaje, pero utiliza convenciones que son conocidas principalmente por los programadores de lenguaje C, C++, C#, Java, JavaScript, Perl, Python, etc. (*json.org, 2016*)

1.12.1 Características

Object en un objeto JSON está formado por una colección de pares de nombre/valor que se encuentra dentro de llaves como se muestra en la figura 6-1.

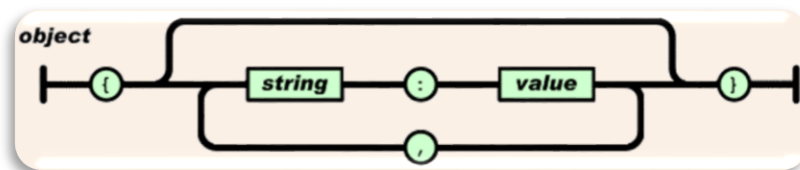


Figura 6-1: Object en un objeto JSON.

Fuente: (*json.org, 2016*)

Un array en un objeto JSON es una colección de valores separados por comas (,) y agrupados dentro de corchetes ([]), como se muestra en la figura 7-1.

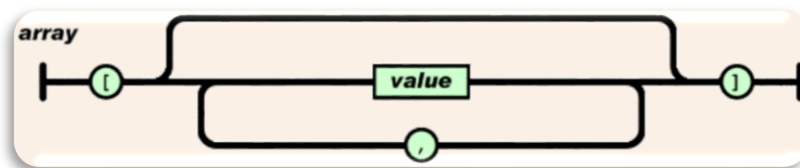


Figura 7-1: Array en un objeto JSON.

Fuente: (*json.org, 2016*)

1.12.2 Ventajas

- JSON es un lenguaje ideal para intercambio de datos.
- JSON es un formato de texto completamente independiente del lenguaje

- Utiliza convenciones que son conocidas por programadores en la mayoría de lenguajes.(json.org, 2016)

1.13 Interfaz de Usuario

La interfaz de usuario es el principal y más importante componente de una aplicación web, ya que permite interactuar con el usuario, la misma debe ser de fácil entendimiento y usabilidad. En el modelo MVC la vista trabaja conjuntamente con el controlador que se encarga de procesar los datos que van ser visualizados de forma personalizada.

Para los desarrolladores comúnmente, es difícil obtener un diseño agradable sin un profesional en el área, sin embargo, se han diseñado diferentes frameworks para que este propósito sea más fácil de alcanzar como **Bootstrap** desarrollado y mantenido por la gigante Twitter Inc. que facilita el diseño al código semántico generado. (Rodríguez, 2012).

1.13.1. Twitter Bootstrap

Es un framework que combina las capacidades de una hoja de estilos CSS con JavaScript, permitiendo crear vistas que se adapten al tamaño de pantalla dependiendo del dispositivo donde se las visualice (responsive design) de una manera sencilla en comparación a realizarlo únicamente con HTML, CSS Y JavaScript. (Rodríguez, 2012)

1.13.2. Características

- Permite la creación de interfaces que se adapten al tamaño de pantalla en base al dispositivo utilizado.
- Integración con las principales librerías de Javascript como JQuery.
- Ofrece un diseño sólido usando LESS (Holas de estilo dinámicas) y estándares como CSS3 y HTML5.
- Es un framework ligero que se integra de manera limpia al proyecto.
- Funciona con todos los navegadores de internet más populares (Mozilla Firefox, google Chrome).
- Dispone de distintos layout predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos. (Rodríguez, 2012)

1.14 Metodología TDD

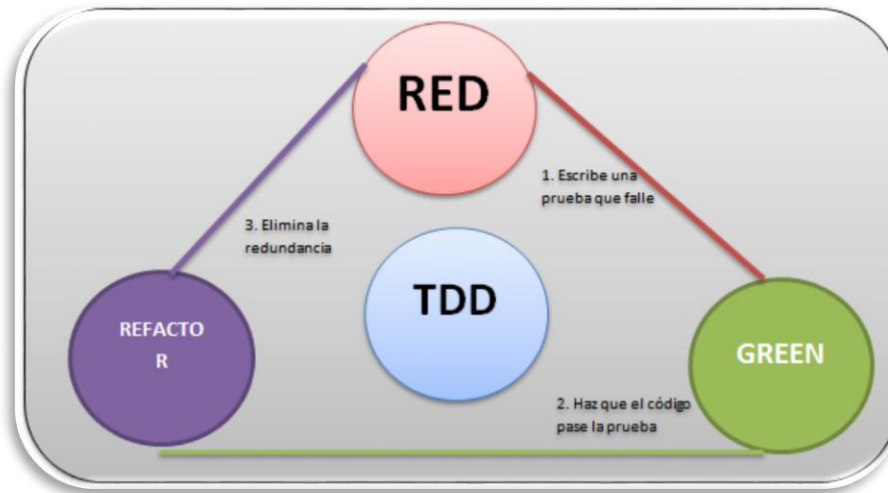


Figura 8-1: Fases Metodología TDD.

Fuente: <http://goo.gl/m3cpEM>

Como muestra la Figura 8-1 de TDD o traducido al español como Desarrollo Guiado por Pruebas, es una metodología ágil extraída del eXtreme Programming para el desarrollo de software, la cual está dividida en tres fases: Test, codificar y refactorizar, cuyo objetivo principal es la de presentar un código robusto, seguro y mantenible, logrando un desarrollo rápido. (Roberto Quesada, 2013)

1.14.1. Características

TDD, puede ser aplicado a dos niveles.

Nivel de micro – iteraciones

- Rápidamente agregar una prueba.
- Correr todas las pruebas y comprobar que una sola falla.
- Hacer pequeños cambios.
- Correr todas las pruebas y comprobar que pasan satisfactoriamente.

Nivel de iteración o Funcional (ATDD)

- El usuario especifica pruebas antes que las funcionalidades sean implementadas.
- Cada vez que el código es escrito, la prueba sirve como criterio de aceptación.

1.14.2. Ventajas

- Escribir la menor cantidad de código posible, puesto que solo se debe escribir el código necesario, para satisfacer las pruebas diseñadas.
- Favorecer que el diseño de las clases, cumpla con los principios de alta cohesión y bajo acoplamiento.
- Garantizar la calidad del software que se produce.
- Minimizar el esfuerzo de los desarrolladores, puesto que el esfuerzo de codificación se centra en escribir solo el código necesario para satisfacer las pruebas, sin distraerse en otros temas.
- Propende por la obtención de un código de calidad, limpio, claro, cada pasada dentro del proceso de refactorización, debe ir acompañada de un refinamiento del código y del algoritmo que se está implementando.
- Reducir a cero el uso de las herramientas de debug.
- Proporcionar una retroalimentación temprana, para la detección y corrección, de errores de codificación y diseño (Ocampo, 2013).

1.14.1. Desventajas

- Es una técnica que requiere disciplina por parte de quien decide aplicarla, y es muy fácil saltarla, o no aplicarla con la debida rigurosidad.
- Para ciertas funcionalidades, es muy difícil hacer un set de pruebas suficiente, que garantice que todos los casos posibles están cubiertos.
- En el caso de las interfaces de usuario, esta técnica no es de fácil aplicación, puesto que el diseño de estas pruebas, requiere de tiempo y herramientas especiales, y la idea de TDD como apoyo a las metodologías ágiles, es propender por ciclos cortos de desarrollo y pruebas.(Ocampo, 2013)

CAPÍTULO II

2. MARCO METODOLÓGICO

El sistema web denominado “**HEMEROTECA VIRTUAL INPC**” representa un proyecto de desarrollo de software con muchas aristas para iniciar su implementación; esto por no ser fundamentado en reglamentos que estructuren su lógica de negocios, es decir, ¿el cómo y que debe desarrollarse?

Esto influye en la propuesta de una guía metodológica ágil, ideal para este tipo de proyecto, garantiza un producto final consistente en su funcionalidad, con un diseño guiado por pruebas de desarrollo en su implementación y contemplando todos los requerimientos para la puesta en producción.

2.1. Fases de la Metodología

2.1.1. *Preparación del Proyecto*

El desarrollo dirigido por pruebas es una práctica de implementación y diseño de software a manera ágil, que se sustenta en tres pilares base.

- El desarrollo de los requerimientos en su justa esencia que el cliente necesita y no más.
- La disminución de defectos de desarrollo de software que lleguen hasta la fase de producción.
- La producción de software modular, el aprovechamiento de su flexibilidad para el cambio en sus requerimientos en todas sus fases provocando reusabilidad.

Con este antecedente el proyecto se determina mediante la recolección y análisis de las necesidades del cliente denominados requerimientos funcionales, dando como resultado un listado sobre lo que debe hacer el software, este se puede evidenciar en la tabla 2 que describe el comportamiento para el sistema informático.

También infiere en groso modo, temas de arquitectura, análisis y diseño de la base de datos de forma incremental, herramientas software para el desarrollo, todo esto mediante reuniones planificadas antes, durante y después de la implementación del sistema.

2.1.2. *Personas y Roles del Proyecto*

El desarrollo del proyecto cuenta con la participación de un equipo de trabajo conformado por 3 personas cuyo rol se especifica a continuación en la Tabla 1-2.

Tabla 1-2: Personas y Roles.

Persona	Contacto	Rol
Ing. Geovanny Silva en representación del Instituto Nacional de Patrimonio Cultural.	gsilva@correo.com	Cliente Representante
Dr. Byron Vaca	bvaca@esPOCH.edu.ec	Coordinador del trabajo de titulación
Sr. Saúl Ibarra	saulibarra@gmail.com	Desarrollador

Realizado Por: Saúl Ibarra. 2017

2.1.3. *Desarrollo dirigido por test de aceptación.*

Los test de aceptación son el criterio que describe al software cumpliendo con los requisitos que el cliente demanda, en esta metodología se puede considerar ejemplos que describen los dueños del producto.

Historias de usuario

Estas historias de usuario son similares a los casos de uso considerando ciertas distancias entre ellas, sin embargo, no intentan describir el requisito como lo contaría su contraparte, señalan con ejemplos ilustrativos y transmite las ideas sin complicaciones en la interpretación.

En TDD cada historia de usuario contiene una lista de ejemplos que describen lo que desea el cliente, con nula ambigüedad, siendo éste una frase simple contada en el lenguaje humano de alrededor de 5 a 10 palabras que resumen que hay que hacer.

Ejemplo:

- Autenticación de usuarios en el sistema.
- Administración de los préstamos de los libros de INPC Regional 3.

Como resultado del análisis de las reuniones y entrevistas se plantea 37 historias de usuarios, entre los que se detalla 3 técnicos, necesarios para el desarrollo del sistema web, así como 34 requerimientos funcionales.

La prioridad de los requerimientos se establece según el criterio de importancia del cliente, esta se la puede representar mediante un rango de 1 a 100 teniendo en cuenta que más cercano a 100, más relevante es la HU, y debe priorizarse para el desarrollo.

En la tabla 2 se describe el Product Backlog conformado por cuatro columnas, donde la primera detalla un identificador de la historia ya sea técnica (HT) o de usuario (HU), la segunda representa la descripción de la historia de usuario, la tercera manifiesta el esfuerzo de cada historia (1 punto de historia oportunamente equivale a 1 día laborable), y esta a su vez se puede estimar de acuerdo a la técnica planning poker (póker de planificación) como base.

Esta es una medida de tamaño relativo de todas las historias respecto a sí mismas, es decir, se puede expresar de manera fácil “La HT02 es más grande que la HT01 y la HT04”. Ahora tenemos el tamaño total del proyecto estimado en una unidad llamada puntos de historias. Estas definidas en libertad por el desarrollador en base a su información, experiencia y auto-organización que le brinda TDD.

Por último, la cuarta columna, que describe la prioridad de la historia de usuario por parte del product owner.

El desarrollo del proyecto tuvo una duración de 6 meses desde la primera reunión con los coordinadores hasta la ejecución del sistema en los servidores del instituto. A continuación, se detallará las iteraciones para la realización del sistema (Tabla 2-2).

Tabla 2-2: Historias de Usuario.

ID	Descripción	Esfuerzo	Prioridad
HT-01	Diseñar el modelo (entidad relación, lógico, físico).	5	90
HT-02	Ingresar datos a la base de datos del módulo del proyecto	1	95
HU-01	Ingresar áreas o departamento a la base de datos.	3	70
HU-02	Modificar áreas o departamento a la base de datos.	5	80
HU-03	Como administrador necesito gestionar paralelos como unidades de estudio.	3	70
HU-04	Ingresar tipo Adquisición a la base de datos.	3	60
HU-05	Modificar tipo Adquisición a la base de datos.	5	50
HU-06	Eliminar tipo Adquisición a la base de datos.	5	55

HU-07	Ingresar tipo Obras a la base de datos.	1	50
HU-08	Modificar tipo Obras a la base de datos.	2	50
HU-09	Eliminar tipo Obras a la base de datos.	3	60
HU-10	Ingresar Tipos Clientes a la base de datos.	5	50
HU-11	Modificar Tipos Clientes a la base de datos.	3	45
HU-12	Ingresar Clientes a la base de datos.	3	45
HU-13	Modificar datos de Clientes a la base de datos.	3	45
HU-14	Eliminar Clientes.	3	45
HU-15	Ingresar Proveedores.	3	45
HU-16	Modificar Proveedores.	2	45
HU-17	Eliminar Proveedores.	2	30
HU-18	Ingresar Tipos Estado a la base de datos.	2	40
HU-19	Modificar Tipos Clientes a la base de datos.	2	50
HU-20	Ingresar Obras a la base de datos.	2	50
HU-21	Modificar Obras a la base de datos.	3	50
HU-22	Ingresar Obras con sus copias ingreso de isbn a la base de datos.	2	40
HU-23	Realizar prestaciones de obras	1	30
HU-24	Realizar donaciones de obras	2	35
HU-25	Realizar registrar devoluciones de obras de obras	2	30
HU-26	Consultar obras disponibles para ser prestadas.	5	25
HU-27	Loguearme en el sistema.	1	20
HU-28	Ingresar usuarios del sistema	1	20
HU-29	Validar datos de ingreso en el sistema.	1	20
HU-30	Seguridad en las páginas de acceso	1	20
HU-31	Visualizar una página publica para consultas de obras	3	20
HU-32	Reportes de las obras prestada.	5	20
HU-33	Reporte de las obras donadas.	5	15
HU-34	Reporte de los clientes con las obras prestadas, áreas y proveedores.	1	15
HU-35	Reporte de los clientes con las obras áreas.	1	15
HU-36	Reporte de los clientes con las obras proveedores.	1	15
HT-37	Como coordinador necesito ver la aplicación desplegada en el servidor del Instituto	1	15

Realizado Por: Saúl Ibarra. 2017

2.1.4. Tipos y Roles de Usuario

En el desarrollo del proyecto se estableció dos tipos de usuarios, sus privilegios en el rol designado se los describe en la Tabla 3-2.

Tabla 3-2: Tipos y Roles.

Tipo de Usuario	Rol
Administrador	<ul style="list-style-type: none">• Agregar, modificar y eliminar información de libros, revistas y demás patrimonios bibliográficos.• Gestionar usuarios del sistema.• Generar reportes de donaciones y préstamos de libros.• Administrar préstamos y devoluciones de las obras literarias.
Usuario	<ul style="list-style-type: none">• Consulta de libros disponibles.• Gestiona reportes de las obras literarias.• Coordina préstamos y devoluciones de las obras literarias.

Realizado Por: Saúl Ibarra. 2017

2.1.5. Recursos de hardware

- Servidor centos 7
- Procesador Core i7 con 3.0 GHz
- Memoria RAM de 8 GB
- Disco duro 1 TB

2.1.6. Planificación con TDD

La planificación de un proyecto de software es una estimación de tiempo, costos y recursos necesarios para el desarrollo e implementación de un sistema. Se realizan diferentes reuniones de retroalimentación para la planificación de diferentes actividades de desarrollo, esto permite realizar un seguimiento al proyecto por parte del cliente, contemplando modificaciones razonables que no recaigan en inconvenientes durante el desarrollo.

Las reuniones con el cliente permitieron establecer las necesidades funcionales para el desarrollo del sistema web con bloques cortos para la ejecución del proyecto como iteraciones de duración de 2 a 4 semanas.

El objetivo de dividir el proyecto en bloques es generar varios entregables durante el tiempo de desarrollo, lo cual le permite al cliente una visión del avance del proyecto y realizar correcciones necesarias. Para el desarrollo del proyecto se consideró a 8 horas diarias como tiempo de trabajo en

días laborables con un total de 36 semanas teniendo como fecha de inicio 01/07/2015 y fecha de finalización 31/03/2016.

Sin embargo, existe diferentes resultados en cada reunión y planificación, uno de ellos es la aparición de nuevos requerimientos y características más severas, estas se han incluido en el product backlog. A todas las reuniones asistieron el product owner y el desarrollador dando dos personas en cada reunión de review como se muestra en la Tabla 4-2.

Tabla 4-2: Product Backlog.

ITERACIÓN	DESCRIPCIÓN	FECHA INICIO	FECHA FIN
Iteración 1	HT1 - Diseñar el modelo (entidad relación, lógico, físico). HT2 - Ingresar datos a la base de datos del módulo del proyecto	18/09/2015	30/09/2015
Iteración 2	HU1 - Ingresar áreas o departamento a la base de datos. HU2 - Modificar áreas o departamento a la base de datos. HU3 - Eliminar áreas o departamento a la base de datos. HU4 - Ingresar tipo Adquisición a la base de datos. HU5 - Modificar tipo Adquisición a la base de datos. HU6 - Eliminar tipo Adquisición a la base de datos. HU7 - Ingresar tipo Obras a la base de datos. HU8 - Modificar tipo Obras a la base de datos. HU9 - Eliminar tipo Obras a la base de datos.	01/10/2015	10/11/2015
Iteración 3	HU10 - Ingresar Tipos Clientes a la base de datos HU11 - Modificar Tipos Clientes a la base de datos HU12 - Ingresar Clientes a la base de datos HU13 - Modificar datos de Clientes a la base de datos HU14 - Eliminar Clientes. HU15 - Ingresar Proveedores	13/11/2015	25/11/2015

	HU16 - Modificar Proveedores. HU17 - Eliminar Proveedores.		
Iteración 4	HU18 - Ingresar Tipos Estado HU19 - Modificar Tipos Clientes a la base de datos HU20 - Ingresar Obras literarias HU21 - Listar las obras literarias. HU22 - Modificar Obras literarias	27/11/2015	01/04/2016
Iteración 5	HU23 - Realizar prestaciones de obras HU24 - Realizar donaciones de obras. HU25 - Realizar registrar devoluciones de obras HU26 - Consultar obras disponibles para ser prestadas.	07/01/2016	28/01/2016
Iteración 6	HU27 - Loguearme en el sistema. HU28 - Ingresar usuarios del sistema. HU29 - Validar datos de ingreso en el sistema. HU30 - Seguridad en las páginas de acceso. HU31 - Visualizar una página publica para consultas de obras.	01/02/2016	15/02/2016
Iteración 7	HU32 - Reportes de las obras prestadas. HU33 - Reporte de las obras donadas. HU34 - Reporte de los clientes con las obras prestadas. HU35 - Reportes de las áreas. HU36 - Reportes de los proveedores	17/02/2016	29/02/2016
Iteración 8	HU37 - Como coordinador necesito ver la aplicación desplegada en el servidor del instituto.	01/03/2016	15/03/2016

Realizado Por: Saúl Ibarra. 2017

2.1.7. Desarrollo con TDD

2.1.7.1. Historias de técnicas preliminares

Se especifica las historias técnicas y de usuario para realizar un levantamiento de información, validación y diseño de la aplicación que se pretende desarrollar. A continuación, se detalla la historia técnica para el análisis y diseño de la aplicación web.

2.1.7.2. *Diseño de base de datos.*

En cada historia técnica o de usuario TDD provoca una serie de preguntas acerca de los diferentes contextos en los que se puede encontrar, aunque en connotaciones tan técnicas como el diseño de la base de datos es más complicado que la historia para un desarrollo neto.

- ¿Cuán cambiante es el diseño del BD?
- ¿Se está preparado para realizar cambios en este momento?
- ¿Si cambia los tipos de datos?

La solución a estas preguntas son afirmaciones los cuales se transforman en test de aceptación.

En la Tabla 5-2, se muestra el test de aceptación de la pregunta ¿Cuán cambiante es el diseño de la BD?

Tabla 5-2: Pregunta 1.

Pregunta	Afirmaciones
¿Cuán cambiante es el diseño de la BD?	Considerar roles y permisos de uso del sistema. Si falta un rol se pueda añadirlo.
	Si se considera un módulo más al sistema. Nuevo módulo debe ser dividiendo en esquemas.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 6-2, se muestra el test de aceptación de la pregunta ¿Se está preparado para realizar cambios en este momento?

Tabla 6-2: Pregunta 2.

Pregunta	Afirmaciones
¿Se está preparado para realizar cambios en este momento?	La base de datos contiene el esquema Entidad Relación debidamente legible.
	Se lo contiene tanto en script debidamente comentado para su fácil cambio.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 7-2, se muestra el test de aceptación de la pregunta ¿Si cambia los tipos de datos?

Tabla 7-2: Pregunta 3.

Pregunta	Afirmaciones
¿Si cambia los tipos de datos?	El tipo de dato fecha debe considerarse en tipo de datos string.
	El tipo de dato int se puede cambiar a dato bigint.

Realizado Por: Saúl Ibarra. 2017

Como conclusión se tiene el diseño de una base de datos debidamente aceptada por el cliente, esquematizada y altamente escalable, refleja un número de 11 tablas necesarias para el alojamiento de la información del sistema web (Figura 9-2).

Tablas como “tipo_usuario”, “usuarios”, “obra”, “estado_obras”, “obras_isbn”, “prestaciones_donaciones” y “proveedor” son de almacenamiento de trato común; es decir, se gestiona todos los datos de los actores principales y relacionándose en su momento generara información mediante su correspondiente proceso en manera habitual en las aplicaciones comunes. Los tipos de dato fuertemente ligados en este sistema se representan datos de uso habitual como datos de tipo entero (int) este el más utilizado para identificadores (PK) y contadores numéricos, las cadenas de texto (varchar) en diferentes longitudes definidas para cada uso en su área, utilizadas mayormente para descripciones, nombres, estados, entre otros textos y en el caso de valores numéricos que se utilizarán para realizar cálculos.

Se utiliza el tipo de dato (float) así como (dateTime) tipo de dato para la asignación de fechas. El diseño de la base de datos se muestra en el manual técnico facilitado en el CD anexo a este documento de titulación, en su ruta “Evidencias Físicas” seguido dentro de “Base de Datos”.

La implantación de la base de datos se dio en 800 KB/500.0 GB, considerando el espacio en disco adecuado y sin representar potenciales inconvenientes cuando la base de datos incremente su tamaño.

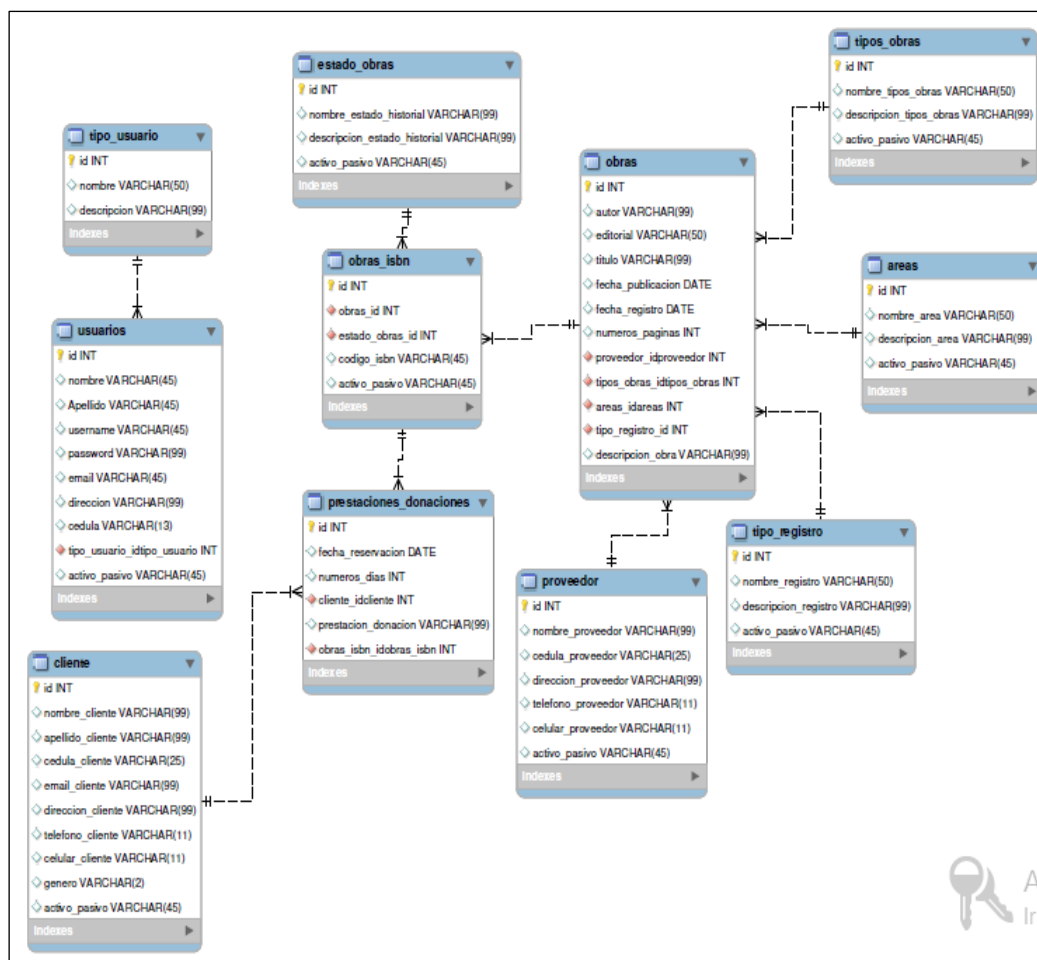


Figura 9-2: Modelo Físico de la BD.
Realizado Por: Saúl Ibarra. 2017

2.1.7.3. Estándar de Codificación

El estilo de codificación en todas sus formas, tanto en la base de datos como en su lógica de programación representa a CamelCase como base para su escritura tanto en palabras solitarias como compuestas, esto se puede comprender mejor a que las palabras escritas en este proyecto se las redacto en forma analógica a la apariencia de la joroba del camello; existiendo dos tipos de CamelCase; UpperCamelCase y lowerCamelCase, considerando esta última como la usada en el proyecto, significando que la primera letra de cada palabra es mayúscula a excepción de la primera palabra en cuya letra inicial es minúscula, como ejemplo primera Palabra Minúscula.

Aplicando la lógica de programación de Laravel y el debido estándar de codificación el desarrollo e implementación del Software de la Hemeroteca para el Instituto de Patrimonio Cultural de la ciudad de Riobamba, tiene la siguiente estructura de desarrollo que se muestra en la Figura 10-2

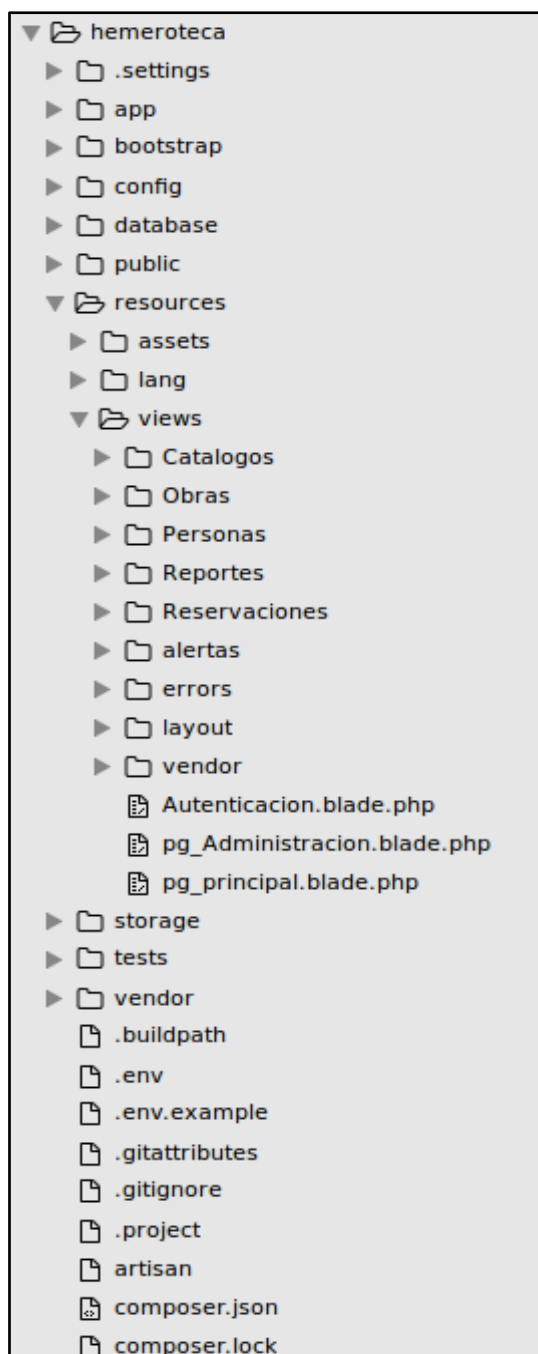


Figura 10-2: Estructura del proyecto - Hemeroteca
Realizado Por: Saúl Ibarra., 2017

2.1.7.4. Desarrollo de las historias de usuario bajo TDD

Las historias de usuarios descritas en la tabla 4 son el resultado de dialogar con el cliente y ayudarlo a describir el requisito en una sola frase. Frases pensando en el negocio del cliente y no en convenciones de desarrollo técnico del sistema.

Una historia de usuario es difícil de estimar, incluso con este formato en lenguaje natural, lo que las hace complejas de describir y ser mejores cada vez que vuelve la iteración, es el proceso evolutivo que llamamos ágil. Para ello tenemos a las diferentes técnicas de TDD que facilita el desarrollo y diseño de software; este consiste en aplicar las siguientes reglas:

1. Describir la historia de usuario a implementar e implantar.
2. Obtener el *Que* se debe hacer y no el *Como* en conjunto con el cliente del software.
3. No escribir código, escribir la prueba de aceptación para la Historia de Usuario.
4. Codificar la prueba para cumplir las pruebas de aceptación.
5. Verificar que la prueba falla.
6. Escribir el código para pasar la prueba.
7. Ejecutar la prueba y comprobar que el código funciona.
8. Refactorizar el código en caso de ser necesario.

Se detalla a continuación el proceso de las 8 reglas implementadas a lo largo del desarrollo del sistema web, tomando como referencia las historias de usuarios con lógica de negocios diferentes y excluyentes una de otra, para tener la perspectiva en cada uno de los casos alternos que pudiera presentarse en el desarrollo ágil guiado por TDD.

Describir la Historia de usuario a implementar e implantar.

HU01.- Autenticación de usuario en el sistema

Obtener el *Que* se debe hacer y no el *Como* en conjunto con el cliente del software.

- ¿Qué sucede cuando ingresa erróneamente las credenciales?
- ¿Qué sucede si no ingresa todos los campos para el login?
- ¿Qué sucede cuando el usuario ingresa correctamente las credenciales?

La solución a estas preguntas son afirmaciones los cuales se transforman en test de aceptación.

En la Tabla 8-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando ingresa erróneamente las credenciales?

Tabla 8-2: Pregunta 4.

Pregunta	Afirmaciones
¿Qué sucede cuando ingresa erróneamente las credenciales?	Se emite un mensaje de credenciales incorrectas.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 9-2, se muestra el test de aceptación de la pregunta ¿Qué sucede si no ingresa todos los campos para el login?

Tabla 9-2: Pregunta 5.

Pregunta	Afirmaciones
¿Qué sucede si no ingresa todos los campos para el login?	Muestra automáticamente un mensaje de campo obligatorio.
	Se enfatiza en color rojo el campo faltante.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 10-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando el usuario ingresa correctamente las credenciales?

Tabla 10-2: Pregunta 6.

Pregunta	Afirmaciones
¿Qué sucede cuando el usuario ingresa correctamente las credenciales?	Permite el ingreso al sistema con su nombre de autenticación en la parte derecha superior.

Realizado Por: Saúl Ibarra. 2017

Codificar la prueba para cumplir los criterios de aceptación

```
<?php
require_once '../src/usuario.php';
class usuario_Test extends PHPUnit_Framework_TestCase
{
    public function testAuntenticacion()
    {
        $this->cb = new usuario();
    }
}
```



```

        $this->assertEquals(1, $this->cb->getUsuario());
    }
}

```

La codificación del test se realiza en favor a resolver la primera prueba de aceptación que se define para la historia de usuario de autenticación.

Verificar que la prueba falla

Esta codificación falla al tratar de crear un usuario y recoger su identificación por medio de una función que no se encuentra en ningún lado.

Esto nos da una apreciación de que recursos, en este caso, clases, atributos y funciones, solamente son necesarios para cumplir con la prueba de aceptación y obtener la Historia de Usuario funcional del sistema ha mediado plazo de forma sólida; es decir, haciendo lo que debe hacer y no más allá con funcionalidades innecesarias.

Escribir el código para pasar la prueba

Se forma entonces el código necesario para satisfacer la necesidad de fallo para la prueba de aceptación, este debe ser el código que irá formándose o modelando con las pruebas realizadas dentro de la guía que nos facilita la metodología TDD.

```

<?php
class usuario
{
    protected $nombre = "Saul Ibarra";
    protected $contrasena = "123456";
    public function getUsuario()
    {
        return $this-> "{usuario: nombre, pass: contrasena }";
    }
}

```

Ejecutar la prueba y comprobar que el código funciona.

El código descrito en el formato anterior hace referencia a la clase “usuario” necesario para obtener un registro nuevo de usuario, junto a sus atributos de “nombre” y “contraseña” las cuales fueron

analizadas en la prueba anterior fallida que demostraron su necesidad absoluta, al igual que el método que devuelve su registro.

Gracias a ello, se puede obtener la primera prueba de aceptación de la HU de autenticación cumplida a cabalidad y solo con las clases, atributos y funciones necesarias para su funcionamiento.

Esto demuestra que el modelamiento del software responde a pruebas que van formando sus necesidades de funcionamiento sin involucrar el código que puede enredar la funcionalidad y no necesaria para cumplir el objetivo planteado como autenticar un usuario al sistema.

Refactorizar el código en caso de ser necesario.

La refactorización de la codificación responde a la necesidad de mejorar el código repetido, esto es necesario cuando se requiera crear varios usuarios para poder realizar la autenticación al sistema; sin embargo, no demuestra necesidad para mejorar y cumplir con el código más eficientemente para la HU autenticación.

El codificar nuevos test para el modelamiento del sistema, la regla de refactorización se vuelve más importante y vital para obtener un software plenamente funcional y limpio en su arquitectura y funcionalidad.

En conclusión, se observa la eficiencia que transmite al desarrollador del sistema, en cuanto visualiza el avance de manera incremental e iterativa con requerimientos cumplidos en función de test de aceptación, que le permite enfocar las necesidades de codificación necesarias, y evita que se realicen funcionalidades que no aportan beneficio alguno al desarrollo del sistema.

Las diferentes funcionalidades obtenidas se las ha implementado de acuerdo a las reglas descritas.

Las Historias de Usuarios 1,2 y 3 se describen para aportar y obtener pruebas de aceptación con diferentes perspectivas las cuales resolver.

HU20.- Ingresar Obras literarias

- ¿Qué pasa si no ingresa todos los campos para registrar la obra literaria?
- ¿Qué sucede cuando ingresa una obra ya registrada?

- ¿Qué sucede cuando ingresa correctamente los datos de la obra?
- ¿Qué sucede cuando necesito ingresar más ejemplares de una misma obra?
- ¿Qué sucede cuando ingreso un ejemplar repetido?
- ¿Qué sucede cuando no tengo uno de los requisitos que el sistema solicita para poder registrar un nuevo ejemplar?

La solución a estas preguntas son afirmaciones los cuales se transforman en test de aceptación.

En la Tabla 11-2, se muestra el test de aceptación de la pregunta ¿Qué pasa si no ingresa todos los campos para registrar la obra literaria?

Tabla 11-2: Pregunta 7.

Pregunta	Afirmaciones
¿Qué pasa si no ingresa todos los campos para registrar la obra literaria?	Muestra automáticamente un mensaje por cada campo de la obra mencionando que dicho campo es obligatorio.
	Se enfatiza en color rojo el campo faltante

Realizado Por: Saúl Ibarra. 2017

En la Tabla 12-2, se muestra el test de aceptación de la ¿Qué sucede cuando ingresa una obra ya registrada?

Tabla 12-2: Pregunta 8.

Pregunta	Afirmaciones
¿Qué sucede cuando ingresa una obra ya registrada?	Muestra automáticamente el mensaje de “nombre de la obra ya ha sido registrado”
	Se enfatiza en color rojo el campo nombre

Realizado Por: Saúl Ibarra. 2017

En la Tabla 13-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando ingresa correctamente los datos de la obra?

Tabla 13-2: Pregunta 9.

Pregunta	Afirmaciones
¿Qué sucede cuando ingresa correctamente los datos de la obra?	Se debe habilitar la opción de ingresar las copias del nuevo ejemplar ingresado.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 14-2, se muestra el test de aceptación de la pregunta ¿Qué pasa si no ingresa todos los campos para registrar la obra literaria?

Tabla 14-2: Pregunta 10.

Pregunta	Afirmaciones
¿Qué sucede cuando necesito ingresar más ejemplares de una misma obra?	La opción de ingresar ejemplares de la obra debe estar a la vista.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 15-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando ingreso un ejemplar repetido?

Tabla 15-2: Pregunta 11.

Pregunta	Afirmaciones
¿Qué sucede cuando ingreso un ejemplar repetido?	Muestra automáticamente un mensaje de la ISBN ya registrada anteriormente.
	Se enfatiza en color rojo el campo faltante

Realizado Por: Saúl Ibarra. 2017

En la Tabla 16-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando no tengo uno de los requisitos que el sistema solicita para poder registrar un nuevo ejemplar?

Tabla 16-2: Pregunta 12.

Pregunta	Afirmaciones
¿Qué sucede cuando no tengo uno de los requisitos que el sistema solicita para poder registrar un nuevo ejemplar?	No permite el registro del nuevo ejemplar.

Realizado Por: Saúl Ibarra. 2017

HU21 - Listar las obras literarias.

- ¿Qué debería listarse?
- ¿Qué opciones debería haber para facilitar el listado de las obras?

La solución a estas preguntas son afirmaciones los cuales se transforman en test de aceptación.

En la Tabla 17-2, se muestra el test de aceptación de la pregunta ¿Qué debería listarse?

Tabla 17-2: Pregunta 13.

Pregunta	Afirmaciones
¿Qué debería listarse?	Todas las obras literarias con el Título, Autor, Tipo, Obra, área.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 18-2, se muestra el test de aceptación de la pregunta ¿Qué opciones debería haber para facilitar el listado de las obras?

Tabla 18-2: Pregunta 14.

Pregunta	Afirmaciones
¿Qué opciones debería haber para facilitar el listado de las obras?	Permitir la opción de visualizar a detalle cada ejemplar listado.
	Permitir editar un ejemplar listado.
	Permitir ver las copias disponibles del ejemplar listado.
	Permitir filtrar el listado de las obras por criterio de buscar.

Realizado Por: Saúl Ibarra. 2017

HU22 - Modificar las obras literarias

- ¿Qué pasa si no ingresa todos los campos para actualizar la obra literaria?
- ¿Qué sucede cuando ingresa correctamente los datos de la obra para actualizar?
- ¿Qué sucede cuando necesito ingresar más ejemplares de una misma obra?
- ¿Qué sucede cuando ingreso un ejemplar repetido?
- ¿Qué sucede cuando no tengo uno de los requisitos que el sistema solicita para poder actualizar un ejemplar?

La solución a estas preguntas son afirmaciones los cuales se transforman en test de aceptación. En la Tabla 19-2, se muestra el test de aceptación de la pregunta ¿Qué pasa si no ingresa todos los campos para actualizar la obra literaria?

Tabla 19-2: Pregunta 15.

Pregunta	Afirmaciones
¿Qué pasa si no ingresa todos los campos para actualizar la obra literaria?	Muestra automáticamente un mensaje por cada campo de la obra mencionando que dicho campo es obligatorio.
	Se enfatiza en color rojo el campo faltante
	Permitir ver las copias disponibles del ejemplar listado.
	Permitir filtrar el listado de las obras por criterio de buscar.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 20-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando ingresa correctamente los datos de la obra para actualizar?

Tabla 20-2: Pregunta 16.

Pregunta	Afirmaciones
¿Qué sucede cuando ingresa correctamente los datos de la obra para actualizar?	Permitir actualizar la obra literaria.
	Muestra un mensaje de confirmación sobre la actualización.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 21-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando ingresa correctamente los datos de la obra para actualizar?

Tabla 21-2: Pregunta 17.

Pregunta	Afirmaciones
	Permitir actualizar la obra literaria.

¿Qué sucede cuando ingresa correctamente los datos de la obra para actualizar?	Muestra un mensaje de confirmación sobre la actualización.
--	--

Realizado Por: Saúl Ibarra. 2017

En la Tabla 22-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando necesito ingresar más ejemplares de una misma obra?

Tabla 22-2: Pregunta 18.

Pregunta	Afirmaciones
¿Qué sucede cuando necesito ingresar más ejemplares de una misma obra?	La opción de ingresar ejemplares de la obra debe estar a la vista.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 23-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando ingreso un ejemplar repetido?

Tabla 23-2: Pregunta 19.

Pregunta	Afirmaciones
¿Qué sucede cuando ingreso un ejemplar repetido?	Muestra automáticamente un mensaje de la ISBN ya registrada anteriormente.
	Se enfatiza en color rojo el campo faltante

Realizado Por: Saúl Ibarra. 2017

En la Tabla 24-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando no tengo uno de los requisitos que el sistema solicita para poder actualizar un ejemplar?

Tabla 24-2: Pregunta 20.

Pregunta	Afirmaciones
¿Qué sucede cuando no tengo uno de los requisitos que el sistema solicita para poder actualizar un ejemplar?	No permite la actualización del ejemplar.
	Muestra un mensaje de error al no actualizar el ejemplar.

Realizado Por: Saúl Ibarra. 2017

HU23 - Realizar prestaciones de obras.

- ¿Qué sucede cuando deseo buscar una obra?
- ¿Qué sucede cuando hay un ejemplar disponible para mostrar?
- ¿Qué sucede cuando no hay un ejemplar disponible para mostrar?

La solución a estas preguntas son afirmaciones los cuales se transforman en test de aceptación.

En la Tabla 25-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando deseo buscar una obra?

Tabla 25-2: Pregunta 21.

Pregunta	Afirmaciones
¿Qué sucede cuando deseo buscar una obra?	Muestra la opción de buscar por título, autor, ISBN y área.
	Filtra la búsqueda a los posibles registros.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 26-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando hay un ejemplar disponible para mostrar?

Tabla 26-2: Pregunta 22.

Pregunta	Afirmaciones
¿Qué sucede cuando hay un ejemplar disponible para mostrar?	Muestra la obra literaria con los ejemplares disponibles.
	Permite buscar y seleccionar al usuario para registrar el préstamo con la fecha actual y los días de préstamo.
	Muestra un mensaje de préstamo exitoso.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 27-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando no hay un ejemplar disponible para mostrar?

Tabla 27-2: Pregunta 23.

Pregunta	Afirmaciones
¿Qué sucede cuando no hay un ejemplar disponible para mostrar?	No muestra ningún ejemplar.
	No permite ninguna acción relacionada al registro de préstamo.

Realizado Por: Saúl Ibarra. 2017

HU25 - Realizar registrar devoluciones de obras

- ¿Qué sucede cuando deseo buscar una obra?
- ¿Qué sucede cuando hay un ejemplar disponible para mostrar?
- ¿Qué sucede cuando no hay un ejemplar disponible para mostrar?
- ¿Qué sucede cuando deseo registrar la devolución?

La solución a estas preguntas son afirmaciones los cuales se transforman en test de aceptación.

En la Tabla 28-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando deseo buscar una obra?

Tabla 28: Pregunta 24.

Pregunta	Afirmaciones
¿Qué sucede cuando deseo buscar una obra?	Muestra la opción de buscar por título y fechas de préstamo.
	Filtra la búsqueda a los posibles registros.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 29-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando hay un ejemplar para devolver?

Tabla 29-2: Pregunta 25.

Pregunta	Afirmaciones
¿Qué sucede cuando hay un ejemplar para devolver?	Muestra la obra literaria.

Realizado Por: Saúl Ibarra. 2017

En la Tabla 30-2, se muestra el test de aceptación de la pregunta ¿Qué sucede cuando no hay un ejemplar disponible para mostrar?

Tabla 30-2: Pregunta 26.

Pregunta	Afirmaciones
¿Qué sucede cuando no hay un ejemplar disponible para mostrar?	No muestra ningún ejemplar.
	No permite ninguna acción relacionada al registro de la devolución.

Realizado Por: Saúl Ibarra. 2017

Las pruebas de aceptación descritas demuestran el cumplimiento de todas las historias de usuario, éstas a su vez guía en el modelamiento con las pruebas codificadas como test unitarios, se realizaron **37** pruebas de aceptación, con un total de **370** pruebas unitarias, las cuales dieron paso al desarrollo eficiente del software para la gestión de la hemeroteca del instituto nacional de Patrimonio Cultural Regional 3.

Las actividades complementarias en el desarrollo del sistema web se dan con la presentación y aceptación de los wireframe para la vista o diseño de interfaces. Estos se obtuvieron de acuerdo a las diferentes entradas y salida de datos para el sistema. A continuación, se presenta los diferentes diseños.

2.1.7.5. Diseño de Interfaces

El proyecto de software para la Hemeroteca cuenta con un diseño de interfaz, una guía visual que representa un esquema ordenado del contenido a mostrar en la aplicación, este no pretende ser una imagen final de un diseño de página web, si no, un conjunto esquematizado de funcionalidad y jerarquía de comportamientos de la aplicación, contribuyendo un buen funcionamiento en conjunto a manera de primera vista para el usuario y terceros implicados en el desarrollo de la aplicación.

Los diferentes Wireframe mostrados a continuación marca una diferenciada visualización entre los diferentes tipos de información tanto de entrada como de salida, cantidad de funciones para su gestión o uso, prioridades, jerarquía y delineamientos para mostrar datos de acuerdo a comportamientos generados dentro de la aplicación sabiendo así el efecto o influencia entre los distintos escenarios en la pantalla de la aplicación uno de esos escenario es el inicio de sesión del sistema (Figura 11-2).



Figura 11-2: Interfaz de inicio de sesión.
Realizado Por: Saúl Ibarra. 2017

En la Figura 12-2, se muestra la pantalla principal del usuario administrador con sus respectivos menús de la aplicación.

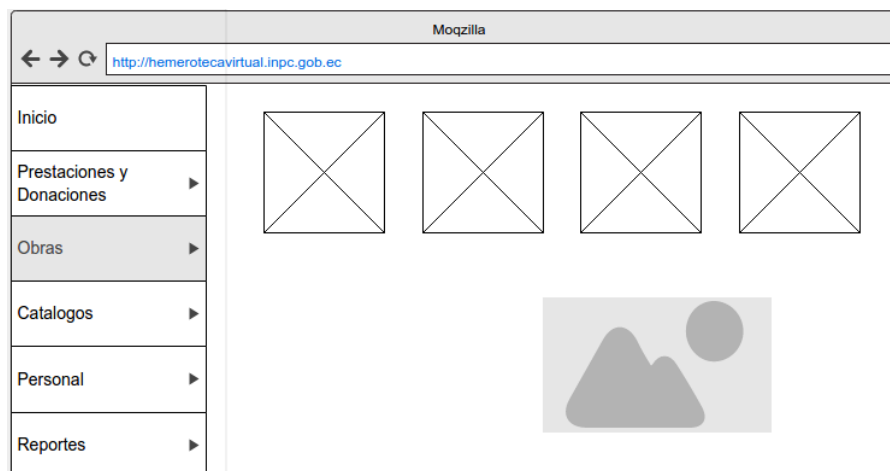


Figura 12-2: Interfaz para el menú de usuario.
Realizado Por: Saúl Ibarra. 2017

En la Figura 13-2, se muestra el bosquejo de la pantalla de la administración de las obras (revistas, libros, folletos).

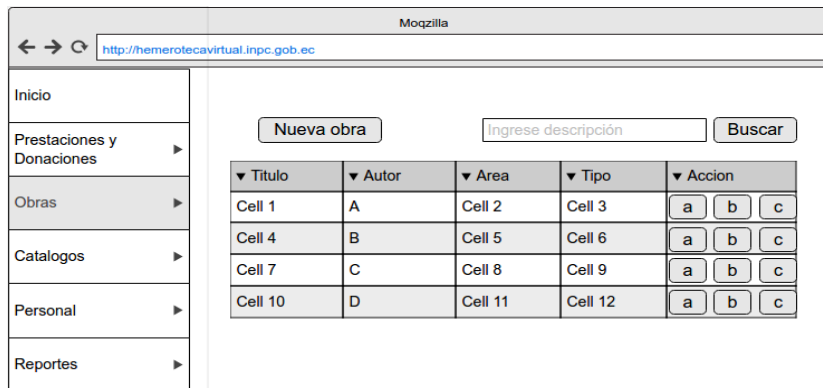


Figura 13-2: Interfaz para la gestión de obras.
Realizado Por: Ibarra S., 2017

En la Figura 14-2, se muestra el bosquejo de los reportes que el sistema va a generar



Figura 14-2: Interfaz para reportes.
Realizado Por: Ibarra S., 2017

El proyecto tuvo una duración de 241 días, con un total de 1928 horas trabajadas, contempladas en la planificación con fecha de inicio 2015-08-03 y de finalización 2016-03-31 para dar paso a la implantación del servidor Centos 7 en la intranet del Instituto Nacional de Patrimonio Cultura de la ciudad de Riobamba.

En el primer sprint en el desarrollo de la base de datos surgieron varios cambios en la estructura de entidades y sus relaciones, por los requerimientos del usuario que se debían adaptar.

En el sprint 6 se tuvo que investigar nuevas funcionalidades del framework que se utilizó para ser aplicada en el sistema, por ejemplo, el requerimiento de seguridad de acceso.

Algunos Sprint se demoraron más de lo estimados por cambios surgidos con las funcionalidades de los requerimientos.

CAPÍTULO III

3. IMPLEMENTACIÓN Y RESULTADOS

3.1. Configuración de servicios.

3.1.1. *Características físicas del servidor.*

- Disco duro 1tb
- RAM 8gb
- Procesador i7
- Monitor 17 pulgadas led

3.1.2. *Instalación del sistema operativo en el servidor físico.*

La Figura 15-3, muestra el inicio del Cd del sistema operativo Centos 7, se introduce el cd en el servidor para su posterior arranque desde el cd.

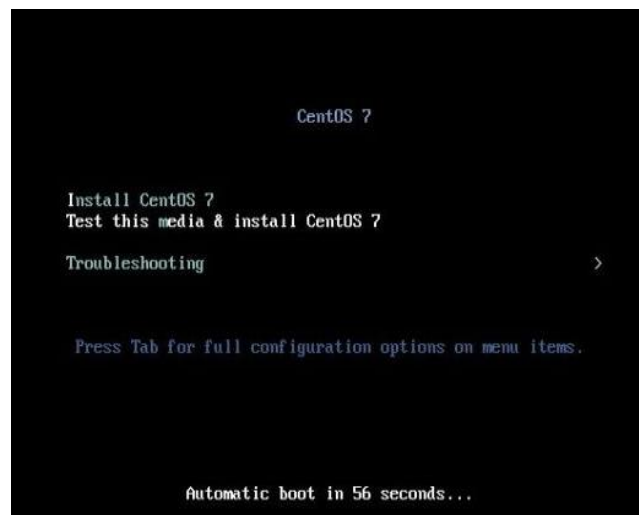


Figura 15-3: Selección del sistema operativo.
Realizado Por: Saúl Ibarra., 2017

Selección del idioma de instalación.

En la Figura 16-3, se presenta la pantalla donde se procederá a escoger el idioma de instalación que se visualizará durante todo el proceso.



Figura 16-3: Selección del idioma.
Realizado Por: Saúl Ibarra., 2017

Configuraciones de opciones de la instalación.

En la Figura 17-3, se presenta la pantalla que nos proporciona algunas configuraciones al momento de la instalación del sistema operativo.



Figura 17-3: Opciones de instalación.
Realizado Por: Saúl Ibarra., 201

Escoger la zona horaria.

En la Figura 18-3, muestra la pantalla que se permite escoger la ubicación de la zona horaria donde nos encontramos ubicado para la configuración de la fecha y hora.



Figura 18-3: Escoger zona horaria.
Realizado Por: Saúl Ibarra., 2017

Configuración del teclado.

En la Figura 19-3, muestra la pantalla que nos permite escoger el idioma para el teclado del servidor.

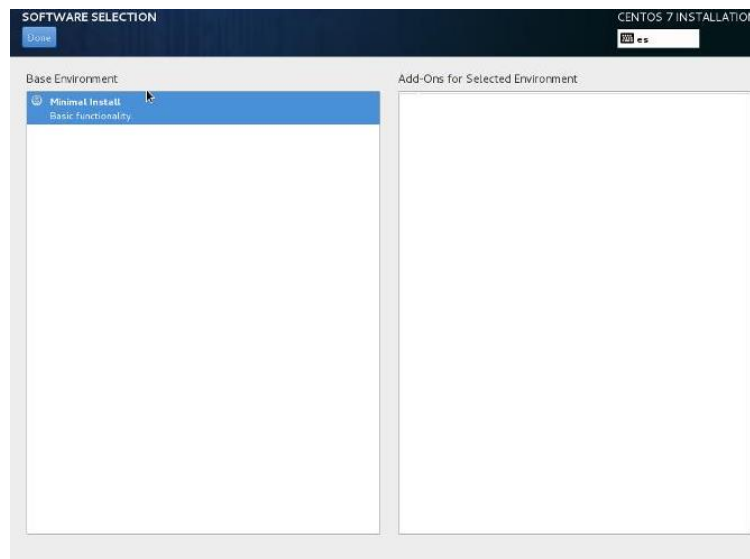


Figura 19-3: Configuración del teclado.
Realizado Por: Saúl Ibarra., 2017

Selección del idioma del sistema.

En la Figura 20-3, se presenta la pantalla donde se visualiza los idiomas donde se debe escoger el idioma del sistema operativo que se desea instalar.

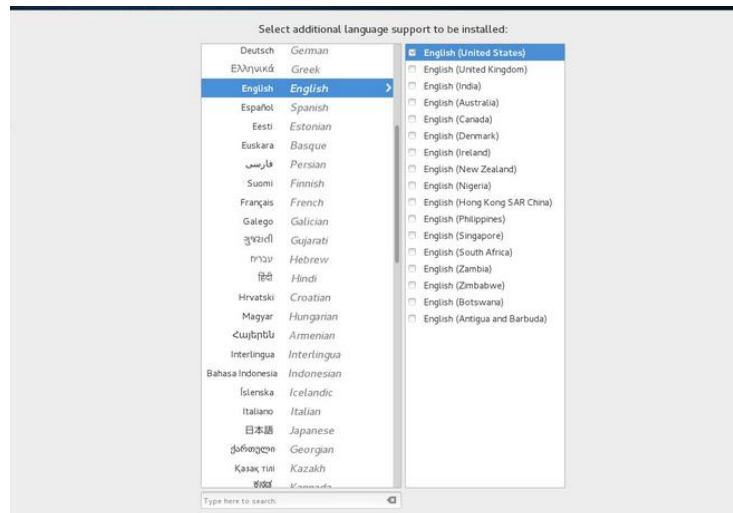


Figura 20-3: Selección del idioma del sistema.

Realizado Por: Saúl Ibarra., 2017

Escoger el tipo de instalación del sistema operativo.

En la Figura 21-3, se presenta la pantalla que permite escoger el tipo de instalación que se desea realizar para el servidor físico.

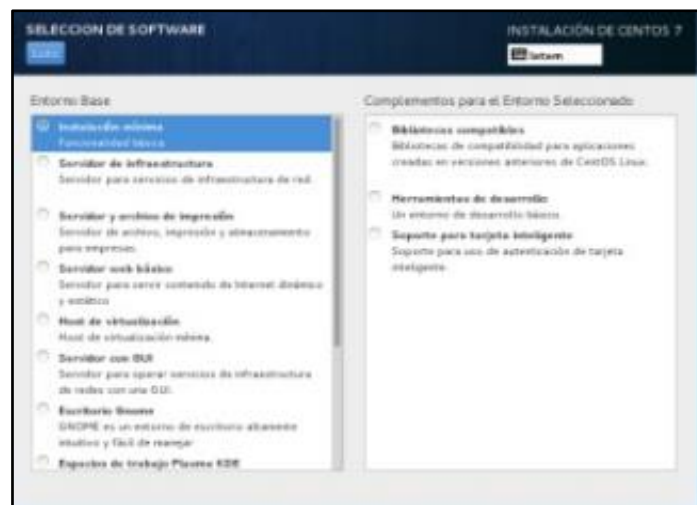


Figura 21-3: Tipo de instalación del sistema operativo.

Realizado Por: Saúl Ibarra., 2017

Configuración de las particiones.

En la Figura 22-3, se muestra la pantalla donde se realiza todas las particiones necesarias que va a tener nuestro servidor al momento se instalar el sistema operativo las principales particiones son:

- /
- swap
- /home
- /boot

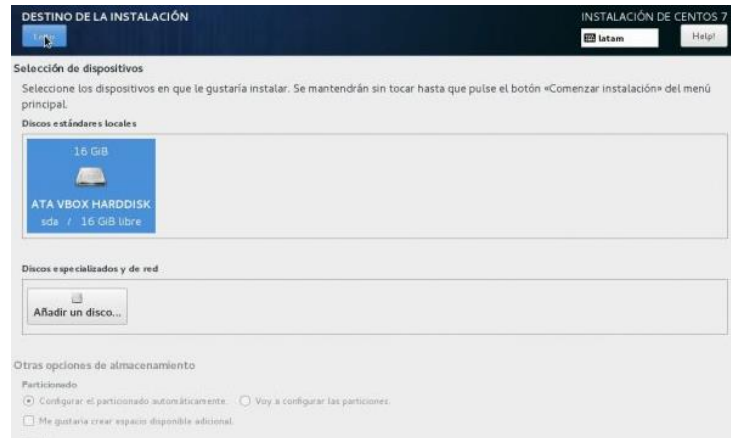


Figura 22-3: Configuración de las particiones.
Realizado Por: Saúl Ibarra., 2017

Montaje de las particiones.

En la Figura 23-3, se muestra la pantalla que permite la creación de las particiones del sistema operativo.



Figura 23-3: Montaje de las particiones
Realizado Por: Saúl Ibarra., 2017

Nuevo punto de montaje.

En la Figura 24-3 se muestra la pantalla la pantalla que permite la configuración de las particiones creadas para la capacidad de espacio en el disco.

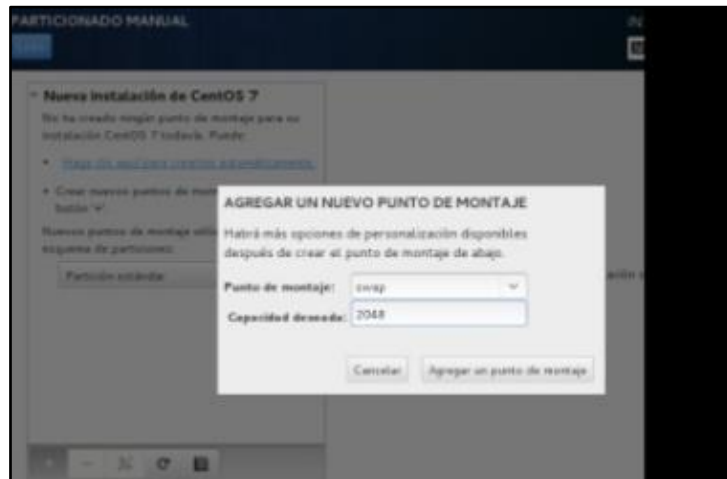


Figura 24-3: Nuevo punto de montaje
Realizado Por: Saúl Ibarra., 2017

Asignación de espacio en disco a las particiones.

En la Figura 25-3, se presenta la pantalla donde se visualiza todas las particiones creadas y la información respectiva de cada una de ellas.

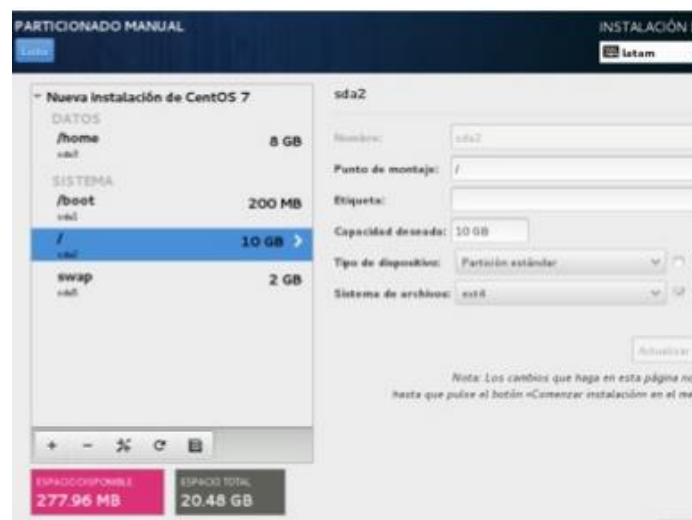


Figura 25-3: Asignación de espacio en disco a las particiones.
Realizado Por: Saúl Ibarra., 2017

Configuración de la red del servidor.

En la Figura 26-3, se muestra la pantalla que permite escoger la opción de la red para su configuración.

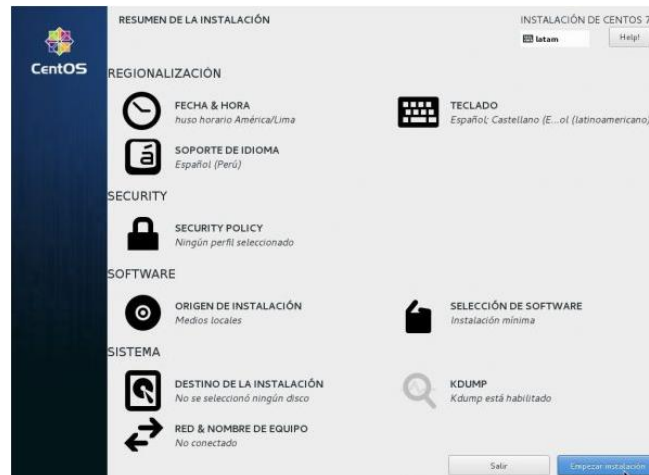


Figura 26-3: Configuración de la red del servidor.
Realizado Por: Saúl Ibarra., 2017

Configuración de la IP estática del servidor.

En la Figura 27-3 se muestra la pantalla donde se configura la red física del servidor donde se coloca una ip estática.

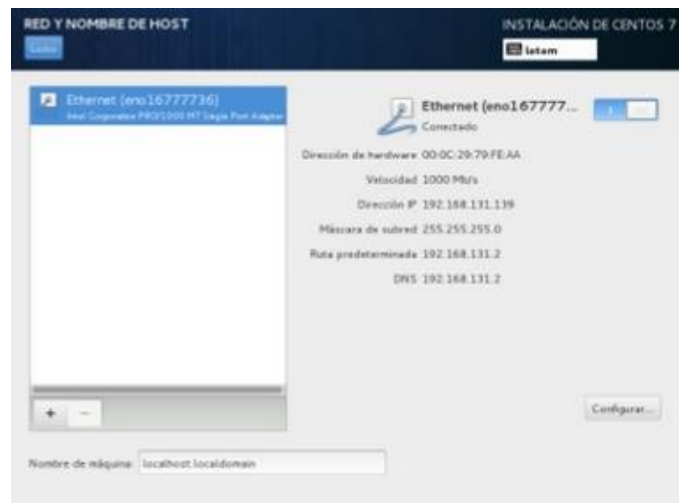


Figura 27-3: Configuración de la IP estática del servidor.
Realizado Por: Saúl Ibarra., 2017

En la Figura 28-3, se muestra la pantalla donde vez realizadas todas las configuraciones se procederá a dar click en el botón **Comenzar instalación**.



Figura 28-3: Inicio de la instalación.
Realizado Por: Saúl Ibarra., 2017

Creación de los usuarios del sistema.

En la Figura 29-3, se muestra la pantalla que permite la creación de los usuarios que el sistema operativo va a tener, por defecto ya viene creado el usuario root.

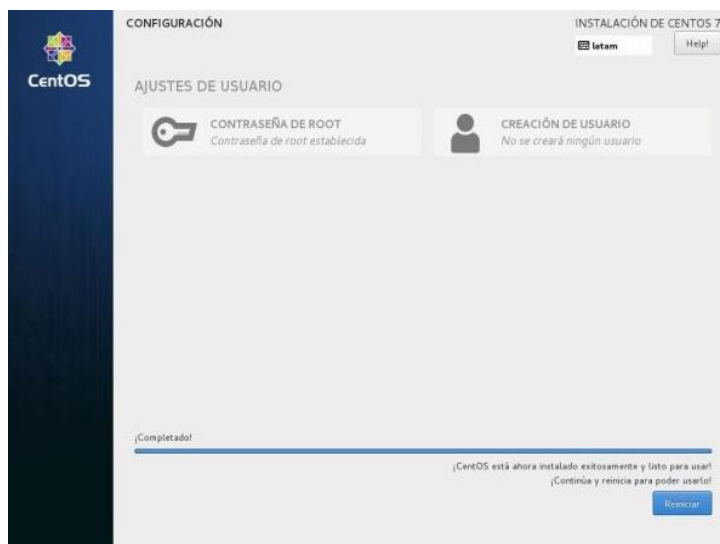


Figura 29-3: Creación de los usuarios del sistema.
Realizado Por: Saúl Ibarra., 2017

Contraseña del súper usuario (root).

En la Figura 30-3 se presenta la pantalla que permite establecer la contraseña del usuario administrador.

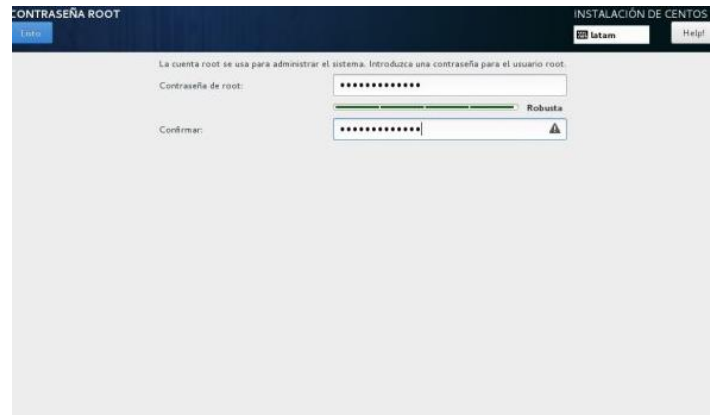


Figura 30-3: Contraseña del root.
Realizado Por: Saúl Ibarra., 2017

Instalación completada.

En la Figura 31-3, se muestra la pantalla donde una vez que la instalación del sistema operativo se ha completado.



Figura 31-3: Instalación completa.
Realizado Por: Saúl Ibarra., 2017

Pantalla de inicio de Centos 7.

En la Figura 32-3 se muestra la pantalla del escritorio del sistema operativo centos7 una vez iniciado sesión.

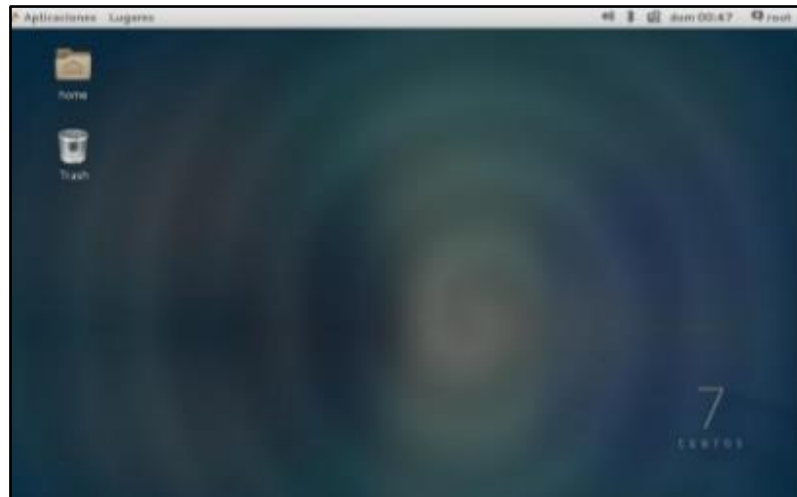


Figura 32-3: Escritorio Centos 7.
Realizado Por: Saúl Ibarra., 2017

3.1.3. *Instalación de Servidor Apache*

Para instalar el servidor apache en Centos 7 se ejecuta el siguiente comando (Figura 33-3).

```
sudo yum install httpd
```

Figura 33-3: Instalación Apache.
Realizado Por: Saúl Ibarra., 2017

Luego de descargar e instalar los paquetes hay que habilitar servicio apache para que arranque por default al encender el servidor e inicializar el servicio (Figura 34-3).

```
systemctl enable httpd.service  
systemctl start httpd.service
```

Figura 34-3: Comandos de configuración Apache.
Realizado Por: Saúl Ibarra., 2017

Al momento de inicializar el servicio se puede verificar si se ha instalado correctamente en la siguiente dirección “localhost” o 127.0.0.1 como se muestra en la Figura 35-3.



Figura 35-3: Comprobación instalación Apache.

Realizado Por: Saúl Ibarra., 2017

3.1.4. Instalación y Configuración de PHP

Se ejecuta la siguiente línea de comando para instalar los paquetes de php5 para apache 2

- `sudo apt-get install libapache2-mod-php5`

La siguiente línea de comando permite conectar desde php con Mysql.

- `sudo apt-get install php5-mysql`

Para alguna configuración adicional de php se ingresa al siguiente archivo.

- `/etc/php5/apache2/php.ini`

Una vez configurado e instalado todo lo necesario, se debe reiniciar el servicio.

- `sudo /etc/init.d/apache restart`

3.1.5. Configuración de Selinux

Se debe configurar el módulo de seguridad del kernel para el despliegue de la aplicación.

- Ir a /etc/selinux/config
- Modificar SELINUX=disabled

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Figura 36-3: Configuración selinux, seguridad del kernel.

Realizado Por: Saúl Ibarra., 2017

Después de reiniciar el sistema se debe verificar la configuración anterior mediante el siguiente comando **getenforce**.

```
~]~ getenforce
Disabled
```

Figura 37-3: Verificar la configuración de selinux.

Realizado Por: Saúl Ibarra., 2017

3.1.6. Configuración de directorios virtuales.

3.1.6.1. Crear la estructura del directorio.

Se necesita es crear la estructura de directorio que mantendrán la información del sitio. El documento raíz (el directorio principal en el cual Apache busca el contenido para mostrar) será configurado en directorios individuales dentro de la ruta /var/www. Crear los directorios aquí para los hosts que se pretende configurar. Dentro de cada uno de estos directorios crear un directorio denominado public_html el cual mantendrá la información pública del sitio y sus respectivos archivos. Por seguridad, para cada uno de los sitios, hay que crear los directorios de la siguiente manera:

```
sudo mkdir -p /var/www/hemeroteca.com/public_html
```


3.1.6.2. *Otorgar permisos*

Si se desea que el usuario regular pueda modificar los archivos en el directorio web, necesita cambiar el propietario haciendo lo siguiente:

```
sudo chown -R $USER:$USER /var/www/hemeroteca.com/public_html
```

La variable \$USER tomará el valor del usuario con el cual actualmente estás identificado. Al hacer esto, el usuario regular ahora es propietario de los directorios public_html donde se almacenará el contenido. Se modifica los permisos para asegurar que el permiso de lectura pueda ser aplicado a archivos y directorios para que las páginas puedan ser desplegadas correctamente:

```
sudo chmod -R 755 /var/www
```

El servidor ahora tiene los permisos necesarios para mostrar el contenido, y el usuario deberá ser capaz de crear contenido en los directorios a medida que sea necesario.

3.1.6.3. *Crear los archivos Virtual Host.*

Los archivos Virtual Host son archivos que contienen información y configuración específica para el dominio y que le indican al servidor Apache como responden a las peticiones de varios dominios. Apache incluye un archivo Virtual Host por defecto denominado 000-default.conf que se puede usar para saltar al punto.

Empezando por copiar el archivo para el primer dominio:

- `sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/hemeroteca.com.conf`
- Abrir el nuevo archivo con el editor como usuario root:
- `sudo nano /etc/apache2/sites-available/ hemeroteca.com.conf`

```
<VirtualHost *:80>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

</VirtualHost>

Personalizar los datos para el dominio y agregar algunas directivas adicionales. Esta sección del Virtual Host coincide cualquier petición que es solicitada al puerto 80, el puerto por defecto de HTTP. Primero, cambiar la directiva ServerAdmin por el correo del administrador del sitio que pueda recibir correos.

- ServerAdmin gsilva@gmail.com

Después de esto, agregar dos directivas. La primera llamada ServerName, que establece la base del dominio que debe coincidir para este Virtual Host. Esto será el dominio. La segunda, llamada ServerAlias, determina nombres futuros que pueden coincidir y servirse como el nombre base o dominio principal. Esto es útil para host tipo www:

- ServerName hemeroteca.com
- ServerAlias www.hemeroteca.com

Lo siguiente por modificar para la configuración es la ubicación del directorio raíz para el dominio.

- DocumentRoot /var/www/hemeroteca.com/public_html

El archivo modificado del Virtual Host debe estar así:

```
<VirtualHost *:80>
    ServerAdmin gsilva@gmail.com
    ServerName hemeroteca.com
    ServerAlias www.hemeroteca.com
    DocumentRoot /var/www/hemeroteca.com/public_html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

3.1.6.4. *Habilitar los nuevos archivos Virtual Host*

Ahora que están creados los archivos virtual hosts, deben ser habilitarlos.

- `sudo a2ensite hemeroteca.com.conf`

Al concluir, reiniciar Apache para asegurar que los cambios surtan efecto:

- `sudo service apache2 restart`

Es necesario recibir un mensaje de información similar a esto:

- Restarting web server apache2

“AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to suppress this message.”

3.1.6.5. *Configurar archivos locales*

Se puede hacer pruebas de funcionalidad de este proceso modificando temporalmente el archivo hosts de la computadora local.

```
sudo nano /etc/hosts
127.0.0.1 localhost
127.0.1.1 guest-desktop
111.111.111.111 hemeroteca.com
```

Esto apuntará directamente cualquier petición para ejemplo.com y pruebas.com en la computadora y enviarlas al servidor en 111.111.111.111. Guardar y cerrar el archivo.

3.1.7.*Configuración de permisos para la carpeta del proyecto*

```
cd /var/www/html
chown apache:apache -R HemerotecaVirtual/find . -type f -exec chmod 644 { } \; && find . -
type d -exec chmod 755 { } \;
```

3.1.8. Instalación de Mysql

En una consola abierta se escribe la siguiente línea de código.

- `sudo yum install mysql-community-server.`

Una vez instalado los paquetes de la instalación, inicializamos el servidor Mysql.

- `sudo systemctl start mysqld.`

Comprobamos si todo fue instalado e inicializado correctamente.

- `sudo systemctl status mysqld`

Configuramos su autoinicio en el arranque.

- `sudo systemctl enable mysqld.service`

Comprobamos su conexión a localhost.

- `mysql -u root`

3.2. Técnicas implementadas para pruebas de usabilidad.

La ISO y la IEC proponen estándares para la evaluación de calidad de diversos productos, siendo uno de ellos, el producto software; donde se evidencia el proceso de valoración en base a parámetros en función de métricas, que aprecia la calidad interna y externa del sistema web.

Para este marco de resultados y su análisis se recoge las bases al procedimiento metodológico que precisa el estándar ISO 9241 pruebas de usabilidad de un sistema, resaltando las características y sub-características a evaluar propuestos en este proyecto de titulación.

El estándar ISO 9241 define a la usabilidad como la medida con la que un producto se puede usar por usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso concreto (Iso, 2000).

En la tabla 31-3 se describe las subcaracterísticas que conforma el modelo base del estándar en miras a la evaluación; que tiene como propósito entender como los usuarios experimenta el sistema web.

Tabla 31-3: Esquema de evaluación.

Característica	Sub-característica	Descripción	Acotación de métrica
Usabilidad	Efectividad	Exactitud e integridad con la que los usuarios alcanzan los objetivos especificados, y por tanto implica la facilidad de aprendizaje, la ausencia de errores del sistema o la facilidad del mismo para ser recordado.	<ul style="list-style-type: none"> • Números de tareas realizadas. • Porcentajes funciones relevantes. • Numero para llamadas de soporte • Porcentaje de usuarios capaces de aprender las características.
	Eficiencia	Recursos empleados (esfuerzo, tiempo, etc.) en relación con la exactitud e integridad con la que los usuarios alcanzan los objetivos especificados	<ul style="list-style-type: none"> • Eficiencia Relativa en comparación con un usuario experto. • Tiempo empleado en el primer intento. • Tiempo productivo. • Tiempo de aprender características.
	Satisfacción	Un factor subjetivo que implica una actitud positiva en el uso del producto	<ul style="list-style-type: none"> • Calificación por parte del usuario. • Frecuencia de reutilización del producto. • Calificación de la facilidad del aprendizaje. • Calificación del tratamiento de errores.

Realizado Por: Saúl Ibarra. 2017

Para la realización de las pruebas de usabilidad se tuvo la guía del “Manual de Técnicas para diseño participativo de interfaces de usuarios de sistemas basados en Software y Hardware” donde podemos encontrar el cuestionario que se aplica preguntas de usabilidad de un sistema.

En el Anexo A, se encuentra el cuestionario de Shneiderman para la evaluación de la usabilidad, este cuestionario está dividido en 7 parte donde las 2 primeras partes trata sobre la información del usuario a encuestar y las 5 partes restantes califica el sistema en usabilidad (Tabla 32-3).

Tabla 32-3: Partes de la encuesta de Shneiderman.

Partes	Descripción
Parte 1	Tipo de sistema a evaluar
Parte 2	Experiencia previa
Parte 3	Reacciones globales del usuario.
Parte 4	Pantalla
Parte 5	Terminología e Información del Sistema
Parte 6	Aprendizaje
Parte 7	Capacidad del sistema

Realizado Por: Saúl Ibarra. 2017

Para el análisis de los datos se utilizó los parámetros de calificación establecido en el cuestionario de evaluación de la usabilidad creado por Shneiderman donde los valores están desde 1 = poco aceptable hasta 9 =aceptable y NA=No Aplicable.

Este cuestionario consta de varias secciones donde cada sección tiene parámetro de calificaciones diferentes, todas estas secciones están orientadas a la efectividad, eficiencia y satisfacción de un sistema.

La Tabla 33-3 muestra la sección Reacciones globales del sistema donde califica de forma general el sistema permitiendo obtener una visión del desempeño y funcionalidad.

Tabla 33-3: Reacciones globales del sistema.

Parámetro	Terrible									Maravilloso
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Frustrante									Satisfactorio
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Aburrido									Estimulante
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Difícil									Fácil
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Potencia inadecuada									Potencia adecuada
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Rígido									Flexible
Valor	1	2	3	4	5	6	7	8	9	NA

Realizado Por: Saúl Ibarra. 2017

La Tabla 34-3 muestra la sección de pantallas donde se califica la navegación dentro del sistema y su fácil entendimiento.

Tabla 34-3: Pantallas.

Parámetro	Difícil de Leer									Fácil de Leer
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	En Absoluto									Mucho
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Confuso									Claro

Realizado Por: Saúl Ibarra. 2017

La tabla 35-3 muestra la sección de terminología e información donde se califica entidad y el contenido del sistema.

Tabla 35-3: Terminología e información del sistema.

Parámetro	Inconsistente									Consistente
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	No Relacionado									Muy bien relacionado
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Confuso									Claro
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	No son de ayuda									Son de ayuda
Valor	1	2	3	4	5	6	7	8	9	NA

Realizado Por: Saúl Ibarra. 2017

La tabla 36-3 muestra la sección de aprendizaje donde se califica la retroalimentación del sistema.

Tabla 36-3: Aprendizaje.

Parámetro	Difícil									Sencillo
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Nunca									Siempre
Valor	1	2	3	4	5	6	7	8	9	NA
Parámetro	Confuso									Claro
Valor	1	2	3	4	5	6	7	8	9	NA

Realizado Por: Saúl Ibarra. 2017

La tabla 37-3 muestra la sección de capacidad donde se califica eficiencia y eficacia del sistema

Tabla 37-3: Capacidad del sistema.

Parámetro	Lento										Rápido
Valor	1	2	3	4	5	6	7	8	9	NA	
Parámetro	Nunca										Siempre
Valor	1	2	3	4	5	6	7	8	9	NA	
Parámetro	Nada Fiable										Fiable
Valor	1	2	3	4	5	6	7	8	9	NA	

Realizado Por: Saúl Ibarra. 2017

Estos parámetros permitirán evaluar de forma más óptima la usabilidad de la aplicación web Hemeroteca Virtual INPC, donde se obtendrán resultado más preciso y confiable.

3.3. Análisis de Resultados

Una vez aplicado el cuestionario para la evaluación de la usabilidad en la aplicación web **Hemeroteca Virtual** dentro del Instituto de Patrimonio Cultural al administrador del sistema se obtuvieron los siguientes resultados.

Análisis de las reacciones globales al sistema.

El sistema tuvo un impacto agradable al usuario administrador por su interfaces amigables y fácil manejo, por el lado de las funcionalidades todos los módulos dentro del sistema realizaban los procesos acordes a sus descripciones.

Es una aplicación altamente adecuada al propósito de prestaciones de obras (Revistas, Folletos, Investigaciones, Libros Culturales).

Análisis de las pantallas.

Los caracteres mostrados en el sistema proveen al usuario de una fácil lectura, así como, resultó de mucha ayuda los aspectos que se destacaron sobre la pantalla como los iconos de cada acción a realizar. El administrador también encontró muy útil y clara la disposición y secuencia de las pantallas respectivamente por los procesos que se realizaban.

Análisis de la terminología y la información del sistema.

El uso de términos en el sistema es consistente y está muy relacionado con el trabajo que se realiza en el proceso de prestación, donación y devolución de las obras. En cuanto a los mensajes mostrados en la pantalla, estos fueron apreciados como consistentes y claros, permitiendo al usuario mantenerse siempre informado de la ejecución de las funcionalidades del sistema y advertido de errores que puedan ocasionarse como: Credenciales incorrectas al momento de logearse en el sistema, Validaciones de los campos al momento de ingresar un registro.

Análisis de la capacidad del sistema.

El sistema fue percibido como un sistema robusto al momento de realizar los procesos más esenciales como la prestación, donación y devoluciones de obras, así también como la generación de los respectivos reportes contemplados por el instituto.

El sistema fue percibido como un sistema rápido al momento de realizar consultas y peticiones de los procesos ejecutados por parte del usuario administrador.

Es notable la reducción del tiempo en el proceso de las prestaciones de obras mediante la utilización del sistema.

CONCLUSIONES

- El estudio y comprensión del Framework de programación web Laravel y el gestor de bases de datos MySQL, permiten el desarrollo de aplicaciones web robustas para ambientes exigentes, donde la información es procesada por procedimientos y procesos de implementación rápidos y seguros.
- El desarrollo de la aplicación web con la metodología TDD para cada requerimiento del INPC, involucra todas las fases de desarrollo de software, aunque exige primero probar para finalizar el desarrollo de las historias de usuarios, es una técnica que genera resultados precisos en poco tiempo con pocos recursos, asegurando la calidad del software.
- La configuración de todos los servicios necesarios para el funcionamiento de la aplicación web Hemeroteca Virtual INPC, involucra desde la configuración precisa y correcta del servidor de aplicaciones web, servidor de base de datos, herramientas de desarrollo para el ambiente web donde va a estar alojado el software construido.
- Las respectivas pruebas de aceptación realizadas por el administrador a la aplicación web **HEMEROTECA VIRTUAL INPC** validan su usabilidad en el aplicativo web con los parámetros de exigencias establecidos.

RECOMENDACIONES

- Al ser una tecnología que va incorporándose ligeramente en el desarrollo de aplicaciones web exigentes el Framework Laravel exige un estudio minucioso y una comprensión técnica avanzada de su funcionamiento e integración con otras herramientas de trabajo (motor de base de datos), ya que la estructura de un proyecto debe respetar y regirse a un esquema organizativo jerárquico, además, las instrucciones de código son diferentes a las de otros Frameworks.
- Las historias de usuario de la aplicación web **HEMEROTECA VIRTUAL INPC** deben ser entendidas correctamente por el desarrollador de software ya que al aplicar las pruebas que exige la metodología TDD en cada requerimiento mientras se implementa el software, se debe obtener el resultado esperado por el cliente.
- La configuración de todos los servicios para el funcionamiento de la aplicación web **HEMEROTECA VIRTUAL INPC**, debe abarcar el levantamiento del servidor de aplicaciones, del servidor base de datos y cuando se realice el levantamiento de debe conocer a donde está orientado sus configuraciones, tener las precauciones técnicas necesarias, respaldos de la información, particiones definidas de manera correcta, para que la aplicación funcione.
- Las respectivas pruebas de aceptación de las aplicaciones web deben estar orientado a las características que se desea evaluar sobre sistema, utilizando normas o estándares ya establecidos.

BIBLIOGRAFÍA

- 1 **AZAUSTRE, C.** *¿Qué es la autenticación con Token?* [En línea]. 2015. [Consulta: 27 Mayo 2016]. Disponible en: <https://carlosazaustre.es/blog/que-es-la-autenticacion-con-token>
- 2 **BAEZ, S.** *Sistemas Web* [En línea]. 2012. [Consulta: 14 Junio 2016]. Disponible en: <http://www.knowdo.org/knowledge/39-sistemas-web>
- 3 **BASALO, A. & ALVAREZ, M. A.** *Qué es AngularJS* [En línea]. 2014. [Consulta: 27 Mayo 2016]. Disponible en: <http://www.desarrolloweb.com/articulos/que-es-angularjs-descripcion-framework-javascript-conceptos.html>
- 4 **CASTRELO CID, A.** *MMO de navegador en tiempo real con Node.js y WebSockets* [En línea]. 2014. [Consulta: 27 Mayo 2016]. Disponible en: <http://hdl.handle.net/2445/59452>
- 5 **CORP, O.** *Mysql database provides full transactional support* [En línea]. 2002. [Consulta: 15 Mayo 2016]. Disponible en: <http://search.proquest.com/docview/206694766?accountid=36724>
- 6 **EGUILUZ, J.** *Introducción a JavaScript* [En línea]. 2006. [Consulta: 27 Mayo 2016]. Disponible en: <https://librosweb.es/libro/javascript/>
- 7 **FLORÍA CORTÉS, A.** *Manual de Tecnicas para el Diseño Participativo de Interfaces de Usuario de Sistemas basados en Software y Hardware* [En línea]. Zaragoza, España. 2001. [Consulta: Marzo 25 2017]. Disponiblen:[http://www.disenomovil.mobi/multimedia_un/01_intro_ux/Manual de Tecnicas para el Diseno Participativo-usabilidad corregido.pdf](http://www.disenomovil.mobi/multimedia_un/01_intro_ux/Manual_de_Tecnicas_para_el_Disenio_Participativo-usabilidad_corregido.pdf)
- 8 **GARY ANTHES, Arlington, VA.** "HTML5 leads a web revolution. Communications of the ACM" [En línea]. *Magazine Communications of the ACM*. Volume 55 Issue 7. 2012, pp.16-17. [Consulta: 27 Mayo 2016]. Disponible en: <http://dl.acm.org/citation.cfm?id=2209256>
- 9 **INPC.** *Instituto Nacional de Patrimonio Cultural* [En línea]. 2015. [Consulta: 03 04 2016]. Disponible en: <http://www.inpc.gob.ec/direcciones-regionales/quito-r1-y-r2>

- 10 **ISO 9241.** *Usabilidad para prueba de un sistema.*
- 11 **JAIN, N., MANGAL, P. & MEHTA, D.** *ANGULARJS: A Modern MVC Framework in JavaScript* [En línea]. 2014. [Consulta: 27 Mayo 2016]. Disponible en: <http://www.jgrcs.info/index.php/jgrcs/article/viewFile/952/610>
- 12 **JSON.org.** *Introducción a JSON.* [En línea]. 2006. [Consulta: 27 Mayo 2016]. Disponible en: <http://www.json.org/json-es.html>
- 13 **JSON.org.** *Javascript Object Notation* [En línea]. 2016. [Consulta: 15 Mayo 2016]. Disponible en: <http://json.org>
- 14 **LARA, W.** *Cómo funciona la metodología de trabajo Scrum* [Blog]. 2015. [Consulta: 27 Mayo 2016]. Disponible en: <https://platzi.com/blog/guia-scrum/>
- 15 **MOYANO MEJÍA, G. W.** *Desarrollo de un sistema web para administración de eventos y control de entrada y salida de empleados. Caso de estudio: Centro de Formación Social" Bethania"* [En línea]. (Tesis doctoral). Pontificia Universidad Católica del Ecuador. Quito - Ecuador. 2015. [Consulta: 27 Mayo 2016]. Disponible en: <http://repositorio.puce.edu.ec/handle/22000/8105>
- 16 **NÚÑEZ, J.** *Expressjs, un framework para Nodejs.* [En línea]. 2013. [Consulta: 27 Mayo 2016]. Disponible en: <http://www.solucionex.com/blog/expressjs-un-framework-para-nodejs>
- 17 **PÉREZ, F.** *Aplicaciones de alto rendimiento con Node.js* [En línea]. 2014. [Consulta: 14 Junio 2106]. Disponible en: <http://www.sidertia.com/Home/Community/Blog/2014/11/24/Aplicaciones-de-alto-rendimiento-con-Nodejs-I>
- 18 **VIVIR, P. D. B.** *Plan nacional del buen vivir* [En línea]. 2013. [Consulta: 15 Mayo 2016]. Disponible en: <http://www.buenvivir.gob.ec/objetivo-11.-asegurar-la-soberania-y-eficiencia-de-los-sectores-estrategicos-para-la-transformacion-industrial-y-tecnologica#tabs2>

ANEXOS

Anexo A: Cuestionario de Shneiderman para la evaluación de la usabilidad

Número de Identificación:

Edad: **Sexo:** **Masculino** **Femenino**

1 Tipo de sistema a evaluar

1.1 Nombre del Hardware	
1.2 Nombre del Software	
1.3 ¿Durante cuánto tiempo ha trabajado con este sistema?	
Menos de 1 hora	
Entre 1 hora y menos de 1 día	
Entre 1 día y menos de 1 semana	
Entre 1 día y menos de 1 mes	
Entre 1 mes y menos de 6 meses	
Entre 6 meses y menos de 1 año	
Entre 1 año y menos de 2 años	
Entre 2 años y menos de 3 años	
Tres años o más	
1.4. En promedio, ¿cuánto tiempo emplea semanalmente con este sistema?	
Menos de 1 hora	
Entre 1 y 4 horas	
Entre 4 y 10 horas	
Más de 10 horas	

2 Experiencia previa

2.1. ¿Con cuántos tipos distintos de sistemas de computadoras ha trabajado?	
Ninguno	
1	
2	
3-4	
5-6	
Más de 6	
2.2. De los siguientes dispositivos, software y sistemas, señalar los que haya utilizado y aquellos 60 con los que esté familiarizado	
Teclado	
Teclado numérico	
Ratón	
Lápiz óptico	
Pantalla táctil	
Joy Stick	
Editor de texto	
Procesador de texto	
Gestor de ficheros	
Ordenador personal	
Diskettes	
Discos duros	
Discos compactos	

3 Reacciones globales del usuario.

Rodee, por favor, los números que reflejan de forma más apropiada sus impresiones sobre el uso de este sistema. NA= No Aplicable.
Hay espacio disponible en la última página para sus comentarios escritos.

3.1 Reacciones globales al sistema									
Terrible									Maravilloso
1	2	3	4	5	6	7	8	9	NA
Frustrante									Satisfactorio

1	2	3	4	5	6	7	8	9	NA
Aburrido					Estimulante				
1	2	3	4	5	6	7	8	9	NA
Difícil					Fácil				
1	2	3	4	5	6	7	8	9	NA
Potencia inadecuada					Potencia adecuada				
1	2	3	4	5	6	7	8	9	NA
Rígido					Flexible				
1	2	3	4	5	6	7	8	9	NA

4 Pantalla.

4.1. Caracteres sobre la pantalla									
Difícil de leer					Fácil de leer				
1	2	3	4	5	6	7	8	9	NA
4.2. ¿Resultaron de ayuda los aspectos que se destacaban sobre la pantalla?									
En absoluto					Mucho				
1	2	3	4	5	6	7	8	9	NA
4.3. ¿Resultó de ayuda la disposición de las pantallas?									
En absoluto					Mucho				
1	2	3	4	5	6	7	8	9	NA
4.4. Secuencia de pantallas									
Confuso					Claro				
1	2	3	4	5	6	7	8	9	NA

5 Terminología e Información del Sistema

5.1. Uso de términos a través del sistema									
Inconsistente					Consistente				
1	2	3	4	5	6	7	8	9	NA
5.2. ¿Se relaciona la terminología de forma apropiada con el trabajo que se realiza?									
No relacionado					Muy bien relacionado				

1	2	3	4	5	6	7	8	9	NA
5.3. Mensajes que aparecen en pantalla									
Inconsistente					Consistente				
1	2	3	4	5	6	7	8	9	NA
5.4. Mensajes que aparecen en pantalla									
Confuso					Claro				
1	2	3	4	5	6	7	8	9	NA
5.5 ¿Le mantiene informado el sistema sobre lo que está ocurriendo?									
Nunca					Siempre				
1	2	3	4	5	6	7	8	9	NA
5.6 Mensajes de error									
No son de ayuda					Son de ayuda				
1	2	3	4	5	6	7	8	9	NA

6 Aprendizaje

6.1 Uso de términos a través del sistema									
Difícil					Sencillo				
1	2	3	4	5	6	7	8	9	NA
6.2. ¿Se relaciona la terminología de forma apropiada con el trabajo que se realiza?									
De forma desalentadora					De forma alentadora				
1	2	3	4	5	6	7	8	9	NA
6.3. Mensajes que aparecen en pantalla									
Difícil					Sencillo				
1	2	3	4	5	6	7	8	9	NA
6.4. Mensajes que aparecen en pantalla									
Nunca					<u>Siempre</u>				
1	2	3	4	5	6	7	8	9	NA
6.5 ¿Le mantiene informado el sistema sobre lo que está ocurriendo?									
Confuso					Claro				
1	2	3	4	5	6	7	8	9	NA

6.6 Mensajes de error									
Confuso					Claro				
1	2	3	4	5	6	7	8	9	NA

7 Capacidad del sistema

7.1. Velocidad del sistema									
Demasiado lento					Demasiado rápido				
1	2	3	4	5	6	7	8	9	NA
7.2 ¿En qué medida es fiable el sistema?									
Nada fiable					Muy fiable				
1	2	3	4	5	6	7	8	9	NA
7.3. El sistema tiende a ser									
Ruidoso					Silencioso				
1	2	3	4	5	6	7	8	9	NA
7.4. Corrección de errores									
Difícil					Sencillo				
1	2	3	4	5	6	7	8	9	NA
7.5. ¿Se tienen en cuenta las necesidades tanto de usuario experimentados como inexperimentados?									
Nunca					Siempre				

8 Comentarios del Usuario

--

